

Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965
PL ISSN 0860 – 1674

Bajtek

MIESIĘCZNY DODATEK DO "SZTANDARU MŁODYCH" NR 9 WRZESIEŃ 1986 CENA 100ZŁ

PORADNIK MŁODEGO PIRATA

GEMINI 10X

C-128

PANAMA JOE

JEDNORĘKI BANDYTA

HOMO INTELIGENS



CZAS ZACZĄĆ BIEC

„Wyśledzić moment historyczny, w którym liczydło dosięgło Rozumu, jest równie trudno, jak ów, co małpę przemienił w człowieka” – tą błyskotliwą uwagą zapożyczoną od Stanisława Lema zainaugurowaliśmy we wrześniu ub.r. wydawanie „Bajtka”. W ciągu roku trafiło do rąk Czytelników 1 mln 700 tysięcy egzemplarzy naszego pisma. Jak przekonuje comiesięczne „głosowanie” odbywające się przy kioskach „Ruchu” – o wiele za mało.

Drugi rok wydawania „Bajtka” rozpoczynamy równocześnie z początkiem nowego roku szkolnego. Jest to dla zawartości naszego tytułu okoliczność istotna gdyż uczniowie, zwłaszcza ze szkół ponadpodstawowych, stanowią znaczną grupę wśród naszych odbiorców (a również i wśród autorów – co nas bardzo cieszy). Dla tych „starych”, zagorzałych czytelników przygotowaliśmy na ten rok sporo niespodzianek. Mamy również nadzieję, że zdobędziemy nowych sympatyków, a wiążemy te oczekiwania z wprowadzeniem do niektórych szkół nowego przedmiotu – „elementy informatyki”. Przedmiot ten wywołuje zresztą mieszane uczucia.

Dzieliłem się już na tym miejscu radością z faktu, że najbardziej powołana ku temu instytucja – czyli szkoła – zaczyna nareszcie wprowadzać uczniów w świat pojęć, metod i urządzeń informatyki. Ale nieśmiałość i ograniczoność tego początku budzi nie skrywany – i to nie tylko przeze mnie – niedosyt. Bo zaczęliśmy wprawdzie w powszechnej edukacji komputerowej iść do przodu, ale inni, myślę o partnerach, z RWPG, zaczęli już w tym czasie biec. W rezultacie dystans między nami stale się powiększa... Zanim więc nasze władze oświatowe zdołają całkowicie się przebudzić, chciałbym przypomnieć co kilka miesięcy temu powiedział w wywiadzie dla „Bajtka” profesor Andriej Jerszow, przyjaciel naszego pisma. Otóż ten najwybitniejszy radziecki informatyk stwierdził bez ogródek:

„Młodzież nie powinna czekać na niczyje decyzje, ani na niczyją zgodę, gdy w grę wchodzi rozwijanie szerokiego programu edukacji informatycznej, której pierwszym etapem powinno być opanowanie umiejętności posługiwania się komputerami osobistymi. Nie trzeba czekać, aż przyjdą one do szkoły. Można organizować kluby komputerowe, można wymyśleć sto innych sposobów...”.

Nie rezygnując z pretensji i wymagań wobec szkoły przejdźmy więc do zasadniczej sfery zainteresowań naszego pisma czyli do społecznego ruchu komputerowego. Jest on naszym oczkiem w głowie jako jedyny, przynajmniej na razie, sposób na to, aby komputery osobiste nie stały się wyłącznie „zabawą dla bogaczy”. Kluby komputerowe stwarzają bowiem szansę kontaktu z komputerem tym, którzy w inny sposób tego kontaktu mieć nie będą. Jednocześnie potencjał pasji działalności społecznej klubowiczów jest tak duży – a udowodnił to chociażby „Abakus”, czy też imponująco rozwijający swoją działalność ursynowski „Maniak” – że grzechem byłoby niepełne jego wykorzystanie. Dlatego, między innymi, zaproponowaliśmy działaczom niektórych klubów prowadzenie rubryk i klanów w „Bajtku”. A już wkrótce ogłosimy współzawodnictwo klubów komputerowych o Złotą Dyskietkę „Bajtka”. Szczegóły za miesiąc.

W drugi rok swego istnienia wchodzi „Bajtek” w doborowym towarzystwie. Otóż ośmielone naszym powodzeniem zaczęły się pojawiać kolejne pisma komputerowe. Również prasa codzienna i tygodniowa zapełniła się rubrykami poświęconymi komputerom osobistym, programowaniu itp. Cieszy nas to poszerzające się „mikro”-towarzystwo, gdyż znaczy to przecież, że rzucone przez nas hasło podjęcia społecznym wysiłkiem powszechnej edukacji informatycznej zostało sformułowane trafnie i na czasie.

Jako pierwszy dołączył do zapoczątkowanego przez „Bajtka” frontu miesięcznik „Komputer”. Zaraz potem pojawił się „IKS” – dodatek do „Żołnierza Wolności”. W momencie, gdy piszę te słowa, podano informację, że bliski jest już czas ukazania się „Mikroklanu”. Usamodzielnia się coraz bardziej „Informik” – stanowiący do tej pory integralną część „Młodego Technika”. Jest coraz lepiej, przynajmniej jeśli chodzi i nakład i ambicje zespołów redakcyjnych. I choć formalnie każde z pism ma nieco inny profil i inaczej wyobraża sobie podstawowy krąg odbiorców, to przecież jednocześnie występuje między nami cicha rywalizacja. Myślę, że jest to zjawisko zdrowe, a Czytelnicy mogą na tej rywalizacji tylko skorzystać.

Ale jest też wiele problemów, które możemy rozwiązać, tylko razem. Dotyczy to na przykład stworzenia i propagowania „kodeksu etycznego” miłośnika informatyki, ochrony programów, utrudniania działalności różnego typu hochsztaplerów, wspólnego redagowania mię-dzywydawniczej „Biblioteczki mikrokomputerowej”, reprezentowania interesów zarówno miłośników mikroinformatyki jak i naszych własnych (być może warto pomyśleć o utworzeniu Klubu Publicystów Informatycznych przy SD PRL), itp. itd. Dlatego redakcja „Bajtka” – najstarszego pisma mikrokomputerowego w Polsce – występuje z inicjatywą zorganizowania spotkania, redaktorów wszystkich pism i rubryk komputerowych. Występujemy z tą inicjatywą publicznie, gdyż nie o prywatne pogaduszki nam przecież chodzi, tylko o omówienie spraw żywo interesujących szerokie kręgi miłośników komputerów osobistych w naszym kraju. Jesteśmy przekonani, że koledzy z bratnich pism i rubryk komputerowych dorzucą swoje pakiety problemów, które należy wspólnie omówić. O wynikach naszej dyskusji poinformujemy obszernie Czytelników. Jesteśmy bowiem przekonani, że czas już najwyższy rozpocząć nowy etap w powszechnej edukacji informatycznej w Polsce. Czas zacząć biec!

Waldemar Siwiński

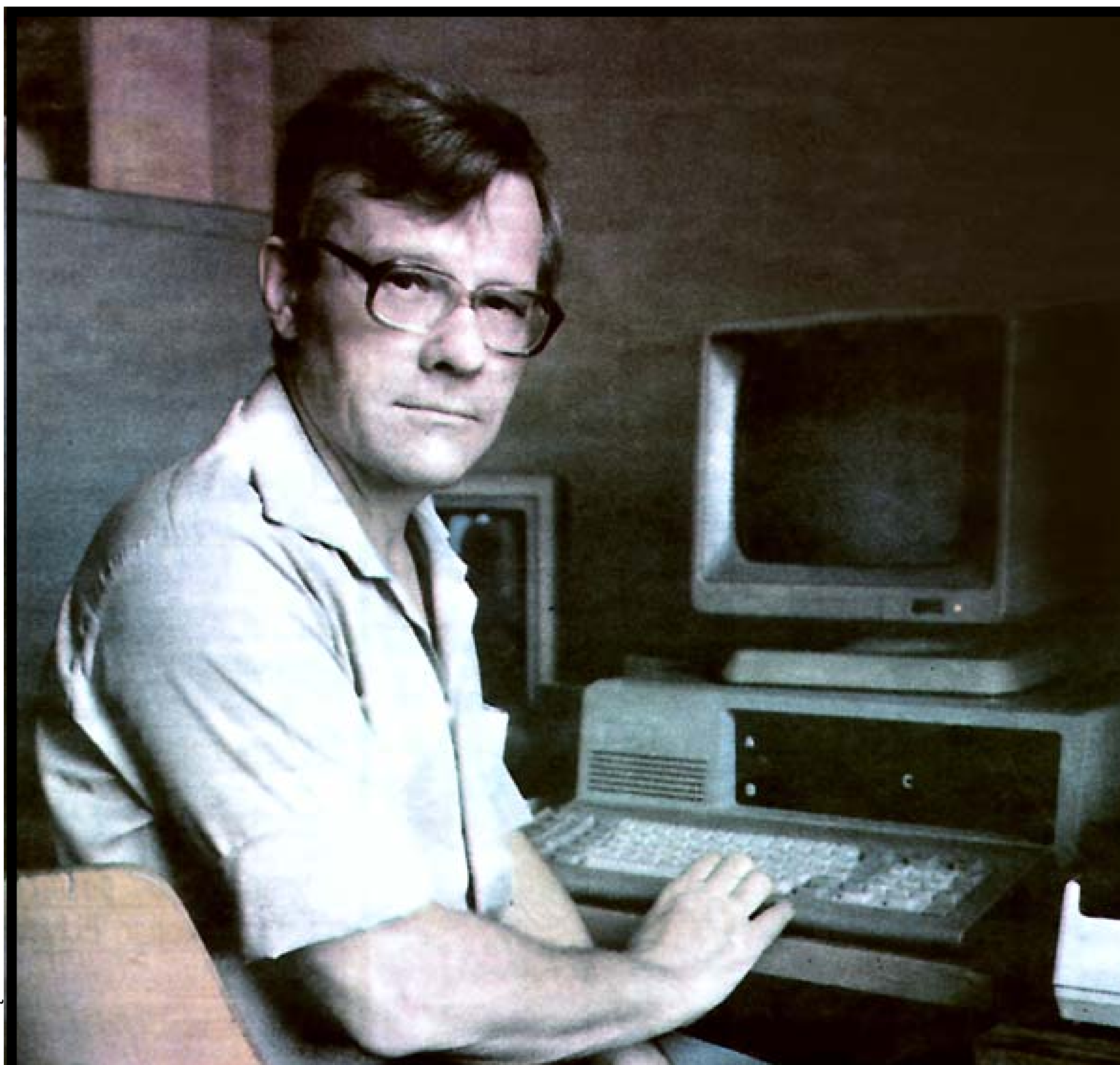


WYBIERZ SAM

„BAJTEK” — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH”.
ADRES: 00-685 Warszawa, ul Wspólna 61. Telefon 21-12-05.
Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych”.
ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM” — kierownik zespołu „Bajtka”), Roman Poznański (z-ca sekretarza redakcji „SM” — sekretarz zespołu „Bajtka”), Krzysztof Czernek, Klaudiusz Dybowski, Sławomir Gajda (red. techniczny), Andrzej Kowalewski, Wiesław Migut, Sławomir Polak. Tomasz Pyć, Wanda Roszkowska (opr. graficzne), Michał Silski, Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski, Sergiusz Wolicki. Zdjęcia w numerze: Leopold Dzikowski i Grzegorz Rogiński.
WYDAWCA: RSW „Prasa-Książka-Ruch” Młodzieżowa Agencja Wydawnicza, a!. Stanów Zjednoczonych 53. 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403,414.
Cena 100 zł.
Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.
Zam. 1149/86, nakład 250 000 egz. P-101.



Bajtek



ZDJĘCIA W NUMERZE: LEOPOLD DZIKOWSKI I GRZEGORZ ROGIŃSKI

START

Rozmowa z doc. dr hab. **Stanisławem Waligórskim**, Członkiem Prezydium Zarządu Głównego PTI, przewodniczącym zespołu, który opracował program przedmiotu "elementy informatyki".

— Od pierwszego września w niektórych szkołach pojawi się nowy uzupełniający przedmiot „elementy informatyki". Przedmiot, którego nauczanie w szkołach praktycznie nie ma żadnych tradycji. Kto jest autorem programu?

— Program został opracowany przez Polskie Towarzystwo Informatyczne na zlecenie Instytutu Programów Szkolnych i zatwierdzony w lipcu 1985 roku przez ministra oświaty i wychowania do realizacji w roku szkolnym 1986/87. Przez pół roku dyskutowaliśmy nad programem, celami nauczania oraz warunkami, które muszą być spełnione, żeby w ogóle rozpoczynanie zajęć miało sens.

— Jakie są cele nauczania?

— Program wyraźnie formułuje cele kształcenia i — na cc zwracam uwagę — wychowania. Oto one: „Zasadniczym celem zajęć z elementów informatyki jest nauczanie metod rozwiązywania za pomocą komputera prostych problemów na poziomie programu liceum ogólnokształcącego, dostosowanych do wiedzy i umiejętności uczniów. W trakcie tych zajęć uczniowie powinni poznać podstawy programowania oraz zdobyć praktyczne umiejętności posługiwania się szkolnym sprzętem informatycznym i jego oprogramowaniem. Dodatkowym celem nauczania elementów informatyki jest stworzenie warunków sprzyjających w korzystaniu z komputera podczas uczenia się matematyki, fizyki, chemii, pracy-techniki i innych przedmiotów".*

Nie jest więc najważniejszym celem nauczanie języków programowania, czy — używając języka młodzieżowego — opowiadanie o bitach latających we wnętrzu komputera. Nie jest także celem dokładne zapoznanie z budową sprzętu. Najważniejsze jest nauczenie, jak posługiwać się komputerem jako narzędziem.

Program został tak pomyślany, aby uczniowie mogli opanowywać zasady samodzielnej pracy z komputerem szkolnym już od pierwszego momentu zajęć. Ćwiczenia z komputerem są ich zasadniczą częścią – niezbędne wiadomości o komputerze, oprogramowaniu, języku programowania i korzystaniu z nich powinny być przekazywane w czasie zajęć praktycznych lub w powiązaniu z nimi. Należy unikać prowadzenia lekcji z tego przed-. miotu metodą wykładów z kredą i tablicą oraz przeładowywania zajęć wiedzą teoretyczną, która nie będzie miała bezpośredniego zastosowania lub związku z zasadniczym celem przedmiotu: nauczenie posługiwania się komputerem w rozwiązywaniu konkretnych problemów. Dlatego zajęcia nie mogą przekształcić się w naukę o komputerach i szczegółach technicznych ich budowy oraz działania. Również nie można z nich robić wyłącznie kursu języka programowania, zwłaszcza bez zwrócenia uwagi na to, jak tym językiem należy się prawidłowo posługiwać i jakie metody programowania trzeba stosować, aby to było rzeczywiście skuteczne.

— Dlatego teoria informatyki została organiczona do minimum?

— Oczywiście, ponieważ nie zamierzamy wychowywać przyszłych profesjonalnych programistów, a użytkowników komputerów. Ważna jest dla nas umiejętność korzystania z już ułożonych programów. Program został tak ułożony, żeby uczeń zaczął pracować z komputerem już w drugiej godzinie zajęć. To jest właśnie nowoczesność informatyki.

— Innym nowum jest to, że do grupy mogą należeć uczniowie z różnych klas: od pierwszej do ostatniej. Czy różnice w poziomie intelektualnym nie utrudnią zajęć?

— Różnice te nie stanowią żadnego problemu, gdyż dużo czasu przeznacza się na pracę indywidualną.

— Wydaje się jednak, że przewidziane na ten cel 20 proc. godzin to niewiele.

— Rzeczywiście, może wydawać się, że nie jest to dużo. Ale przecież w trakcie całych zajęć uczniowie pracują z komputerem. Treść programu celowo jest ujęta ogólnie. Nie napisano, co robić w każdej godzinie — to powinno zależeć od inwencji nauczyciela i od tego, jak będzie współpracować z klasą.

— Czy program konsultowano z nauczycielami?

— Przede wszystkim zespół, który go opracowywał, składał się w większości z nauczycieli-praktyków. Ale oczywiście były prowadzone dyskusje w bardzo szerokim gronie, odwoływaliśmy się do doświadczeń szkół, w których uczono elementów informatyki, wykorzystaliśmy dostępną w kraju literaturę — także zagraniczną.

— Jednym z ulubionych argumentów przeciwników zmian jest stwierdzenie, że najbardziej szkodliwe jest eksperymentowanie na uczniach.

— A czy można w ogóle wyobrazić sobie wprowadzenie nowego przedmiotu bez eksperymentalnego sprawdzenia? Znam kilka szkół we Wrocławiu i w Warszawie — a są także inne w kraju — gdzie program sprawdzono i odbyło się to bez szkody dla uczniów. Zresztą, program cały czas jest sprawdzany. Także najbliższe lata będą jego sprawdzianem.

— Czy będzie modyfikowany i w jakim trybie?

GRA O JUTRO

— Program jest zatwierdzony i chyba przez najbliższe lata nie będzie zmieniany. Co bedzie później? Zależy od doświadczeń, ale trzeba zdawać sobie sprawę, że gwałtowne zmiany — nawet w programie dopiero wprowadzonym — też nie są dobre.

— W rozdziale „Treści kształcenia i wychowania" w punkcie „Obsługa mikrokomputera" czytamy, że jednym z zadań jest „zapoznanie się ze szkolnym mikrokomputerem". A cóż to takiego jest? Przecież praktycznie nie istnieje „polski mikrokomputer szkolny"! Szkoły mają zbieraninę najróżniejszego sprzętu, gromadzonego na zasadzie: bierzemy wszystko co się tylko uda zdobyć.

— Dlatego zakładamy w programie, że jeśli zajęcia się rozpoczynają i jest przewidziane zapoznanie się z mikrokomputerem — to szkoła już ma laboratorium i przypadają nie więcej niż trzy osoby na jedną klawiaturę. Poza tym bardzo dokładnie jest określone, co to takiego szkolny mikrokomputer. Musi mieć pamięć wewnętrzną nie mniejszą niż 64 kilobajty, pamięć zewnętrzna na dyskach miękkich, monitor ekranowy z grafiką (możliwość rysowania kresek), klawiaturę odpowiadającą polskim normom i z polskim alfabetem, możliwość przyłączenia drugiego monitora i drukarki. Pożądana jest pamięć kasetowa lub możliwość przyłączenia magnetofonu kasetowego, urządzenia do lokalizacji punktu na ekranie: mysz, manipulator albo pióro świetlne, struktura otwarta — umożliwiająca dalszą rozbudowę, łącznie z innymi urządzeniami peryferyjnymi lub przyłączenie do sieci:

— Czy w obecnej sytuacji nie są to wymagania zbyt rygorystyczne?

— Inny sprzęt do nauczania elementów informatyki w szkole praktycznie się nie nadaje. Można przy jego pomocy robić różne próby w Kółkach mikrokomputerowych — nie ograniczonych jednak ściśle szkolną metodyką nauczania, oceniania, promowania itd. W przybliżeniu — z wyjątkiem klawiatury i dyskietek — zbliżony do wymagań jest Spectrum. Obecnie przy jego pomocy są prowadzone eksperymentalne zajęcia. Ale jeszcze lepsze byłyby Amstrady czy Macintosh'e — chociażby ze względu na pamięć na dyskach elastycznych. Pamięć na taśmach magnetofonowych jest zbyt powolna i bardzo zawodna co poważnie utrudnia prowadzenie zajęć.

Zajęcia z tego przedmiotu mogą być prowadzone wyłącznie w tych szkołach, które mogą zapewnić wszystkim uczestnikom dostęp do mikrokomputera z oprogramowaniem dostosowanym do realizacji tego programu. Pracę uczniów z komputerem należy traktować jako główny sposób przyswajania wiadomości. Trzeba także zapewnić uczniom dostęp do komputera poza formalnymi godzinami zajęć.

— Jedną z barier jest polska klawiatura?

— Tak i jest to trudny do spełnienia warunek. Chodzi jednak o to, żeby nie uczyć młodzieży posługiwania się żargonem; nie pisać „trojkat" zamiast „trójkąt". Przy pisowni fonetycznej mogą występować błędy ortograficzne — co z punktu widzenia dydaktyki jest niedopuszczalne. Wprowadzenie komputerów nie może powodować kaleczenia języka. Zespół PTI opracował polskie LOGO, przygotowano także jego interpreter. Ministerstwo Oświaty i Wychowania będzie te opracowania rozpowszechniać.

— Doszliśmy do punktu krytycznego. Praktycznie nie istnieje polski komputer szkolny — na razie są tylko jego założenia. Meritum, najbardziej znany rodzimy mikrokomputer personalny nie spełnia tych warunków. Wyprodukowano zaledwie kilka prototypów Juniora — praktycznie więc nie ma o czym mówić. Wszyscy się zgadzają, że czas najwyższy zacząć kształcenie informatyczne. Ale jednocześnie nie ma zaplecza, są opinie, że nie jesteśmy przygotowani do wprowadzenia tego przedmiotu.

— Zgadzam się, dlatego uparcie podkreślam: powodzenie akcji zależy od dobrego wyposażenia i przygotowania nauczycieli. Nie chcę wartościować — co ważniejsze, są to dwa równorzędne i niezbędne czynniki. A wyposażenie to nie tylko komputery i urządzenia peryferyjne, a także oprogramowanie i materiały pomocnicze, chociażby poradniki metodyczne.

— Ministerstwo Oświaty i Wychowania opracowało program rozwoju kształcenia informatycznego. Mówi on o różnych formach: od kółek mikrokomputerowych — a więc zajęć amatorskich — po wprowadzenie nowego przedmiotu do szkół które musi być poprzedzone dwuletnimi przygotowaniami. Przez dwa lata 1986-1987 resort oświaty będzie się przygotowywał do wprowadzenia przedmiotu elementy informatyki. W tym czasie można prowadzić te zajęcia w szkołach które mają już pewne doświadczenia, zaplecze i przygotowanych nauczycieli. Nie ma żadnych naciskόw żeby było tych szkół jak najwięcej. Wręcz przeciwnie, zaleca się jak największą ostrożność. Niestety, dwa lata to bardzo mało. Dlatego w pierwszej kolejności sprzęt powinien być dostarczony do ośrodków kształcenia nauczycieli. Nauczyciel nie może uczyć się informatyki razem z uczniami. Byłoby to wręcz tragiczne w skutkach.

— Dla resortu dwa lata to mało, ale dla uczniów którzy chcą się uczyć informatyki — to bardzo dużo,

— Tak, ale są także inne formy tej edukacji. Poza tym warto zastanowić się nad stwierdzeniem uczniowie chcą się uczyć. Co to znaczy? Po pierwsze nie wszyscy chcą. Po drugie: duża część młodzieży słyszała już jakieś nowinki, czytała artykuły, oglądała programy w telewizji, ale nie miała kontaktu z komputerem. Dlatego tak ważne jest, żeby zetknięcie z komputerem nie spowodowało zniechęcenia do informatyki. Niesłychanie o to łatwo, jeśli zajęcia będą prowadzone niewłaściwie przez osoby nie przygotowane do tego, w tłoku, bez dostępu do komputera, bez szansy poznania, jakie są jego możliwości.

Przedwczesna powszechność i obligatoryjność nauczania informatyki zawiera ogromne ryzyko popełnienia poważnych pomyłek. Rozsądek nakazuje poprzedzić powszechne wprowadzanie elementów informatyki okresem próbnym, w którym gromadzone byłyby doświadczenia w szkołach, posiadających eksperymentalne, właściwie wyposażone, laboratoria z odpowiednim sprzętem i oprogramowaniem, oraz odpowiednio przygotowanych nauczycieli, którzy byliby współtwórcami właściwych metod nauczania, współpracując z odpowiednimi uczelniami, ośrodkami doskonalenia lub placówkami metodycznymi.

— Jaki procent szkół jest w stanie już w tym roku rozpoczynać zajęcia?

— Około 1 proc... Jeżeli uwzględnimy, że jest ok. 3 tys. szkół mogących wprowadzać przedmiot, a kursy przygotowujące objęły 200-300 nauczycieli, jeżeli uwzględniamy brak sprzętu — to otrzymamy ten jeden procent.

— Nie brzmi to zbyt optymistycznie. Producenci sprzętu komputerowego twierdzą wręcz, że nie jesteśmy w stanie produkować więcej niż 5 tys. mikrokomputerów rocznie, i tak jest to liczba solidnie zawyżona. MERA-ELZAB produkuje rocznie 2 tys. Meritum i jest to produkcja uboczna, do której zakład dopłaca.

— Dlatego ministerialny program komputeryzacji szkół zakłada, że należy zapewnić jednostkom resortu oświaty możliwość zakupu 75 tys. sztuk kompletnych zestawów mikrokomputerowych wraz z odpowiednimi urządzeniami peryferyjnymi i materiałami.

— W naszych warunkach liczba 75 tys. wydaje się wręcz astronomiczna. Jaki jest najniższy próg, od którego można by zacząć wprowadzenie przedmiotu?

— Moim zdamem rachunek zaczyna się od 10 tys. sztuk rocznie. Jeżeli przemysł zacznie produkować polski mikrokomputer — a powinien zacząć, żeby wspomniane dwa lata nie poszły na marne — to wtedy będzie o czym mówić. Proponuję jednak nie zastanawiać się nad tym, czy przemysł wyprodukuje — będą to po prostu jałowe dyskusje. Proponuję konkluzję: nowy program nauczania musi być wprowadzony do szkoły poważnie. Młodzież jest bardzo inteligentna i łatwo zauważy, czy traktuje się ją poważnie, czy też są to działania pozorne.

Rozmawiali:
Sławomir Polak
Roman Wojciechowski

Orientacyjny przydział godzin

1.	Obsługa mikrokomputera	2	(3%)
2.	Praktyczne zastosowania mikrokomputera	6	(8%)
3.	Tworzenie rysunków na ekranie	4	(5%)
4.	Procedury	12	(16%)
5.	Styl programowania	12	(16%)
6.	Nieelementarne metody grafiki	12	(16%)
7.	Działania na tekstach	12	(16%)
8.	Ćwiczenia samodzielne	15	(20%)

* Wszystkie cytaty pochodzą z „Programu liceum ogólnokształcącego (profil podstawowy i matematyczno-fizyczny) elementy informatyki. Uzupełniający przedmiot nauczania. Warszawa 1985, WSiP



Sprzęt, którym dysponujemy to Apple IIe z pamięcią 128kB, monitor dwa napędy dysków elastycznych oraz drukarka Imagewriter.

Jakie przyjęliśmy nałożenia przy projektowaniu sposobu wykorzystania naszego Apple'a?

1. Aby system był naprawdę efektywnie wykorzystany trzeba napisać oprogramowanie ściśle dostosowane do naszych potrzeb.

2. Programy muszą być bardzo proste w obsłudze, aby każdy z pracowników wydziału mógł się nimi swobodnie posługiwać.

3. Każdy wydruk związany z jedną z naszych baz danych musi być gotowym dokumentem

4. Staramy się wszystkie możliwe potrzebne listy, zawiadomienia i dokumenty oraz tabelaryczne podsumowania oraz analizy związane z tym tematem zawrzeć w formie gotowych programów wywoływanych jednym klawiszem.

5. Komputer będzie pracował w tematach szczególnie pracochłonnych — oczywiście możliwych do zautomatyzowania.

Myślę, że na pełną prezentację sposobu wykorzystania naszego ko-mputera w pracy potrzebny byłby cały numer „Bajtka". Przedstawię więc tylko parę wybranych tematów.

Warunkiem wyjazdu grupy młodzieży polskiej w formie hufca pracy bądź praktyki zawodowej do CSRS, NRD bądź WRL jest umowa podpisana między KG OHP, a zakładem pracv w jednym z tych trzech krajów. O każdej takiej umowie rejestrowane są podstawowe dane: termin rozpoczęcia i zakończenia pracy, ilość wyjeżdżających dziewcząt i chłopców, resort przedsiębiorstwa zatrudniającego, województwo realizujące daną umowę itp. Pełny zbiór tych danych (wraz z programami) zajmuje prawie 3 pełne dyskietki (po 169 kB każda). Oczywiście nie jest to jedynie martwe archiwum. Dysponujemy wszystkimi potrzebnymi programami do zarządzania tą bazą danych. Są więc programy do edycji i poprawek poszczególnych rekordów (umów). Bogaty zestaw programów umożliwia wydruk wszystkich umów z danego kraju w układzie chronologicznym, według resortów lub według przedsiębiorstw.

Kilka programów służy do tworzenia zestawień i tabel, które pozwalają szczegółowo i pod różnym kątem analizować zatrudnienie młodzieży polskiej za granicą.

W czasie przeglądania około 1000 rekordów drukowane jest wybrane przez użytkownika zestawienie. Cały proces trwa kilka minut. Metodą tradycyjną (albo „nowoczesną” z wykorzystaniem kalkulatora) co najmniej kilka godzin.

Wokół bazy danych zawierającej wszystkie umowy powstało również kilka innych programów np. histogram obrazujący przyjazdy i wyjazdy młodzieży za granicę oraz liczba osób przebywających w danym kraju.

Ostatnio powstał program umożliwiający obliczenie wysokości ubezpieczenia na okres podróży młodzieży do i z miejsca zatrudnienia.

Drugim tematem są Międzynarodowe bozy Pracy odbywające się co roku w kilkunastu krajach świata. Uczestnicy rekrutowani przez Komendy Wojewódzkie OHP, wypełniają ankietę, w której między innymi podają okres, w którym chcieliby wyjechać, preferowany kraj, znane języki. Dane te wraz z danymi osobowymi umieszczane są w pamięci naszego komputera. Następnym zbiorem jest lista obozów pracy ochotniczej w różnych krajach świata. Po zdaniu egzaminu między innymi z języka uczestnicy zostają przydzieleni na obozy.

Odbywa się to automatycznie co zapewnia wysoki stopień zgodności przydziału z preferencjami kandydata. Praktycznie cała korespondencja z uczestnikiem (zawiadomienie o przydziale na obóz, zawiadomienie o odprawie i terminie odbioru paszportu i innych dokumentów niezbędnych do wyjazdu) odbywa się „automatycznie”.

Jednym z istotnych kręgów odbiorców naszych wydruków są organizacje partnerskie za granicą.

Trzecim pakietem programów jest zestaw do obsługi Międzynarodowych Obozów Pracy w Polsce.

Programy są ściśle dostosowane do naszych potrzeb (rezerwacje miejsc dla organizacji, grupowe wysyłanie zaproszeń, lista uczestników wg organizacji wysyłającej, lista uczestników konkretnego MOP-u).

W działalności naszego Wydziału wykorzystujemy również napisane przeze mnie programy. Muszę też przyznać się, że kilka spraw — które próbowaliśmy zautomatyzować — lepiej i po prostu wygodniej prowadzić metodą tradycyjną. Całość napisanego przeze mnie i wykorzystywanego przez nasz Wydział oprogramowania to około 20 tysięcy linii kodu. Zdecydowana większość to **BASIC APPLESOFT i DOS 3.3**. Wybór tego języka i systemu operacyjnego podyktowany był głównie w miarę kompletną dokumentacją w momencie rozpoczynania pracy. Choć zdaję sobie sprawę z faktu, iż przyszłość mikrokomputerów w systemach biurowych to **IBM PC-XT** oraz pakiety typu **LOTUS 1-2-3, Framework, Symphony, dBase III** i sieci lokalne — to jednak nie należy jeszcze przez długi czas lekceważyć potencjalnych pożytków jakie może przynieść w biurze mikrokomputer 8--bitowy, np. **Apple, Amstrad 6128** czy **Commodore 128**

A tak na marginesie to uważam, że relatywnie wysoki koszt sprzętu mikrokomputerowego w Polsce nie uzasadnia stosowania go tylko jako inteligentnej maszyny do pisania, choć niewątpliwie procesor tekstu jest zazwyczaj entuzjastycznie



przyjmowany jako bardzo wygodne narzędzie pracy. Oczywiście z takiego programu również korzystamy.

Niewątpliwie najbardziej doskonałą częścią zestawu jest drukarka **Imagewriter I.** Model ten powstał jako podstawowa drukarka do **Macintosha**. Kilkadziesiąt krojów pisma, bardzo solidna konstrukcja mechaniczna, bufor o pojemności 2 kB i przede wszystkim niezawodność przy intensywnej pracy to jej podstawowe zalety. Do wad zaliczyć trzeba spore koszty eksploatacyjne związane z wymianą taśm barwiących. Mimo, że drukarka nie posiada trybu **NLQ** (**N**ear **L**etter **Q**uality) to jednak odpowiednie połączenie typów druku zapewnia druk korespondencyjny. Również szybkość druku (160 znaków na sekundę) plasuje ją wysoko pośród tanich drukarek mozaikowych.

Napędy dyskietek — choć już dość przestarzałe (pojedyncza gęstość i jednostronne) spisują się znakomicie. Nie najwyższa gęstość zapisu wpływa korzystnie na niezawodność i umożliwia korzystanie z najtańszych dyskietek. Niewątpliwie największą wadą Apple jako systemu do zastosowań biurowych, jest mała poiemność dyskietek. Producent znając ten fakt wypuścił w zeszłym roku 3,5 calowy napęd dysków elastycznych umożliwiający sformatowanie dyskietki 3,5 calowej na 800 kB (dwustronna o podwójnej gęstości).

System operacyjny DOS 3.3 umożliwia zakładanie zarówno plików o dostępie sekwencyjnym jak i o dostępie bezpośrednim. Rozmiar rekordu w pliku o dostępie bezpośrednim nie może przekraczać 256 znaków. Syntaktyka operacji dyskowych jest dość skomplikowana ale rozbudowane funkcje umożliwiają efektywne zarządzanie zbiorami na dysku.

Niewątpliwie największą wadą języka **BASIC Applesoft** z punktu widzenia zarządzania zbiorami rzędu 100— 300 kB jest brak możliwości zwolnienia pamięci zajmowanej przez macierz (odpowiednik komendy **ERASE** a$ w BASICU **Amstrada 6128** — gdzie a$ jest dowolną macierzą tekstową. Receptą jest deklarowanie jak najmniejszych macierzy – co za tym idzie — formowanie danych w pliki o dostępie bezpośrednim. Należy także unikać zbędnych operacji tekstowych. Generalnie plik o dostępie bezpośrednim choć narzuca pewne rygory na wielkość rekordu i zostawia dużo „białych plam” na dyskietce jest znacznie łatwiejszy w dalszym przetwarzaniu. Pliki sekwencyjne stosować należy tylko przy niewielkich zbiorach, które rzadko będą ulegały zmianie. Wszystkie te uwagi odnoszą się również do **Commodore'a 64**, w którym sposób organizacji macierzy w pamięci jest podobny.

Zaprojektowanie struktury danych na dysku to fundament każdego programu do przetwarzania danych. Brać należy pod uwagę nawet częstotliwość wykorzystywania przez użytkownika różnych opcji programu. Opcje wykorzystywane często powinny być szybsze w działaniu. A decyduje o tym w dużym stopniu sposób organizacji danych na dysku. Warto też — jeśli powstać ma kilka różnych baz danych — stworzyć możliwie szeroko sparametryzowany program uniwersalny, który następnie po niewielkich przeróbkach i ustaleniu parametrów stanie się programem do zastosowań szczególnych.

Jerzy Szafranek, lat 28
absolwent Wydziału Matematyki, Mechaniki i Informatyki UW, programista w KG OHP

OCHOTNICZE HUFCE PRACY
KOMENDA GLOWNA
ADRES DLA KORESPONDENCJI
00-920 WARSZAWA ul:NOWY SWIAT 18/20

WYDZIAL HUFCOW ZAGRANICZNYCH
WARSZAWA ul:GRZYBOWSKA 79 tel:32-43-99

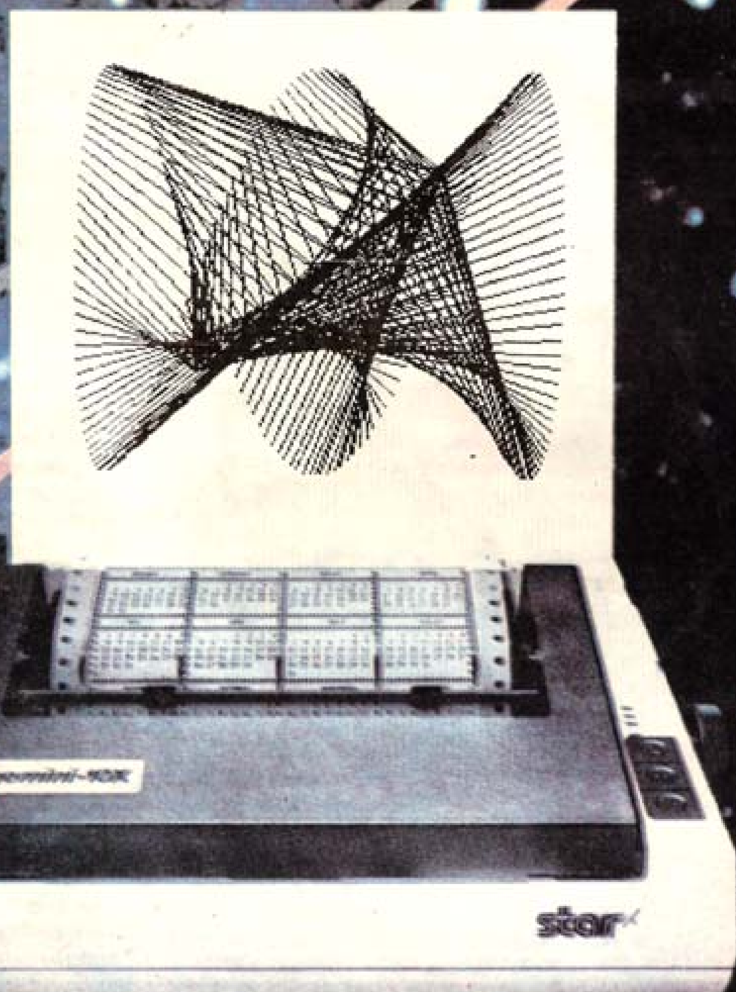
L.dz. /WHZ/86

WARSZAWA 1986.07.15.

STAN ZATRUDNIENIA OHP w CSRS w dniu 2207

KATEGORIE 1 2 3 4

R E S O R T	HUFCE	CHLOPCY	DZIEWCZETA	KADRA	RAZEM
ROLNICTWO	75	982	804	91	1786
BUDOWNICTWO	48	1065	20	49	1085
LASY	30	496	211	36	707
PRZEMYSL	18	290	125	22	415
O G O L E M	171	2833	1160	198	3993



DANE TECHNICZNE

Typ drukarki: mozaikowa
Głowica: 9-cio igłowa (możliwość wymiany)
Prędkość drukowania: 120 znaków na sekundę (przy 10 znakach na cal) dwukierunkowy przesuw głowicy
Matryca znaku: standardowego 9x9
wyrazistego 18x9
drukowanego podwójnie 18x18
graficznego 6x6
Tryby grafiki: niskiej rozdzielczości 60x72
wysokiej rozdzielczości 120x144
super wysokiej rozdzielczości 240x144
Zestaw znaków: 96 standardowych znaków ASCII
96 znaków kursywy (pochylonych)
64 znaki specjalne
32 znaki graficzne
96 znaków programowanych
Rodzaje druku: normalny (10 znaków na cal)
elite (12 znaków na cal)
ścieśniony (17 znaków na cal)
wyrazisty drukowany podwójnie
poszerzony (5; 6; 8,5 znaków na cal)
indeksowy dolny i górny
kursywa
Dodatkowe możliwości: autotest
ciągłe podkreślenie cofanie głowicy
tabulacja pionowa i pozioma, marginesy
interface 7-mio lub 8-mio bitowy
Średni czas pracy głowicy: 10x10 7 znaków
Ilość kolumn druku: znaki normalne — 80, elite – 96, ścieśnione — 136, poszerzone — 40, 48, 68
Odstęp między liniami: 1/6, 1/8, 7/72 cala, oraz programowany n/76 i n/144 cala
Szerokość papieru: z perforacją — 3 do 10 cali (7,5—25 cm) zwijany — do 8,5 cala (21 cm) pojedyncze arkusze — 8 do 10 cali (20—25 cm)
Interface: standardowo — równoległy (Centronics) opcjonalnie — szeregowy (RS-232C)
Taśma barwiąca: na szpuli, szer. 15 mm (jak w maszynie do pisania)
Waga: 15,4 funta (7 kilogramów)

Począwszy od tego numeru ..Bajtka” rozpoczynamy nowy cykl — testy komputerów i urządzeń peryferyjnych. Nie będą to zwykłe opisy technjczne, lecz raczej uwagi z punktu widzenia użytkownika. Testy beda dotyczyły wyłącznie sprzętu, na którym mieliśmy okazję pracować co najmniej kilka miesięcy.

Drukarka **Gemini-10X** produkowana jest w wersjach przystosowanych do większości komputerów domowych i osobistych. Może więc pracować z **Apple II**, **Atari**, całą rodziną **Commodore**, bliźniakami: **Amstradem i Schneiderem, IBM, Osborne, TRS-80**. Przekazana nam do testowania **Gemini-10X** posiada interface **Centronics** i pracuje w naszej redakcji z **Amstradem 6128 i 464** a po zastosowaniu dodatkowego interfaced z **Commodore 128 i Vic 20**.

Gemini-10X należy do grupy drukarek amatorskich. Wyróżania się jednak wśród innych solidnością wykonania i niezawodnością. Na pierwsza rzut oka nie prezentuje się zbyt efektownie. Jest dość duża — konstruktorzy projektując ją nie żałowali materiałów i miejsca, nie wysilali się na miniaturyzację. Wystarczy podnieść pokrywę, żeby zobaczyć, że wewnątrz nie jest wcale ciasno. Jest to drukarka mozaikowa z dziewięcioigłową głowicą. Jej prędkość drukowania wynosi 120 znaków na minutę. Drukarka posiada w swym zestawie znaków m.in. 96 znaków programowanych przez użytkownika i 88 znaków narodowych. Niestety wśród znaków narodowych brak jest polskich liter (ą,ć,ę,ź,ó itd.).

Gemini dysponuje ośmioma rodzajami druku. Poszczególne rodzaje można dodatkowo łączyć. Istnieje także możliwość umieszczania znaków na innej wysokości niż reszta tekstu, a więc wpisywania indeksów, potęg itp.

Gemini pozwala użytkownikowi na zdefiniowanie aż 96 znaków własnych. Można więc stworzyć swój alfabet. Można także dopisać brakujące polskie litery. Jest jednak pewna trudność. Otóż matryca znaku ma co prawda wymiary 9 x 9, jednak szerokość znaku nie może przekraczać 7 punktów. Dwa szeregi punktów na górze lub na dole matrycy muszą być zawsze puste. Jeśli więc projektujemy literę Ą musimy wówczas ogonek przy nózce skierować do góry, bądź obniżyć całą literę o dwa punkty. Ani jeden, ani drugi sposób nie jest niestety elegancki. Wszystka małe litery wchodzą natomiast świetnie.

Drukarka posiada mikroprzełączniki pozwalające ustawić na stale pewne parametry druku. Osiem z nich znajduje się wewnątrz obudowy. Można za ich pomocą wybrać narodowy zestaw znaków, tryb dru kowania, gęstość druku, wielkość przesuwu do nowej linii. Cztery mikroprzełączniki na tylnej ścianie obudowy służą do włączania czujnika końca papieru, auto-

Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X
Test Bajtka - drukarka Gemini 10X

matycznego przeskoku do nowej linii, sterowania buforem pamięci i wyboru interface'u. Dwie sprawy spędzają sen z powiek użytkownikom drukarek, mianowicie papier i taśma barwiąca. Szczególnie zdobycie tej ostatniej jest bardzo kłopotliwe i dość kosztowne. **Gemini-10X** pracuje z taśmę dokładnie taka samą jak w maszynach do pisania. Można więc założyć do niej taśmę produkcji krajowej (należy jednak przewinąć ją na oryginalne szpulki, gdyz te różnią się nieco od naszych i na końce taśmy założyć blaszki powodujące zmianę kierunku obrotów). Lepsze wyniki daje jednak zastosowanie taśm produkcji zachodniej na podłożu z tworzyw sztucznych.

Jeżeli chodzi o papier, to mamy dużą swobodę. Możemy używać papieru perforowanego — drukarka posiada tzw. traktor, który jednak można w każdej chwili odłączyć. Możemy także korzystać z papieru z rolki (np. papier do dalekopisów). Warto dodać, że do drukarki dołączona jest bardzo wygodna podstawka na rolkę z papierem.

Istnieje także możliwość drukowania na pojedyńczych kartkach, gdyż **Gemini** posiada czujniki końca strony.

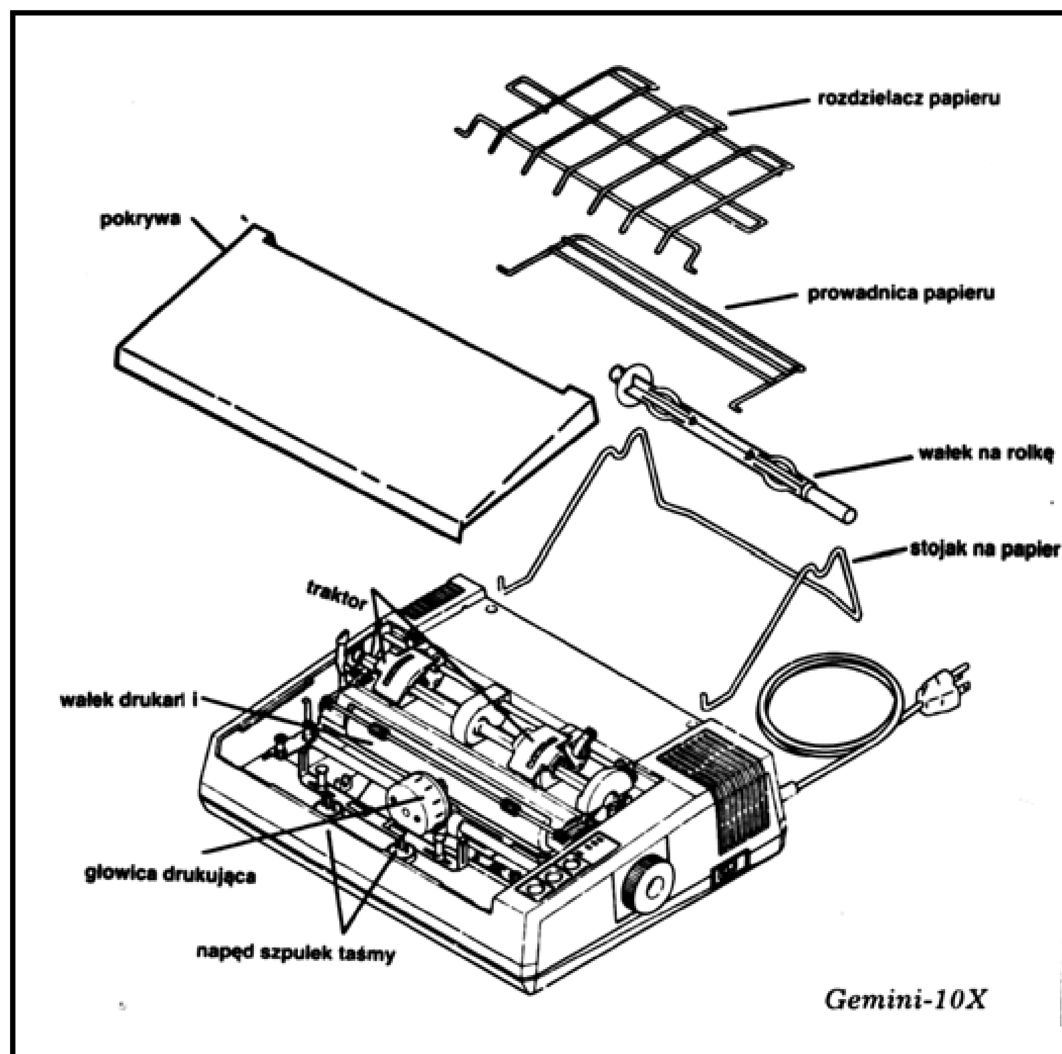
Najwygodniejszy w pracy jest papier z perforacją. W przypadku drukowania dłuższych tekstów czy rysunków na papierze bez perforacji ulega on zwykle przesunięciu na wałku i wydruk wychodzi krzywy.

Należy więc kontrolować co chwilę czy papier przesuwany jest prawidłowo.

Gemini posiada solidna, konstrukcję. Nie zmienia to jednak faktu, ze nadal pozostaje ona urządzeniem niezmiernie delikatnym, które łatwo uszkodzić. Trzeba więc przestrzegać kilka podstawowych zasad (dotyczą one zresztą także innych drukarek). Po pierwsze nie wolno — w żadnym przypadku – łączyć drukarki z komputerem, jeśli którekolwiek z tych urządzeń jest włączone. Jest to równie groźne dla komputera jak i dla drukarki. Nie wolno także wyłączać drukarki podczas pracy ani tez włączać w momencie, gdy komputer... zawiesił się podczas wykonywania instrukcji rukowania. Jeśli drukarka jest włączona przewijanie papieru wykonujemy za pomoca klawisza L.F lub F.F a nie ręcznie. Warto też sprawdzić czy napięcie w sieci nie spada poniżej 200 V albo czy w tej samej sieci nie pracują duże urządzenia elektryczne, np. silniki. W takim przypadku warto kupić stabilizator — opłaci się. Szkodliwy wpływ na pracę drukarki mają również silne wibracje i pył.

Podczas czteromiesięcznej — bardzo intensywnej — eksploatacji w naszej redakcji, **Gemini-10X** pracowała bez najmniejszej awarii. Blisko połowa wydruków programów ukazujących się w „Bajtku" pochodzi z tej właśnie drukarki. Na niej również wykonywane są niektóre ilustracje, np. schematy blokowe. Uważamy, że jest to odpowiednia drukarka do komputerów osobistych i domowych.

Roman Poznański



Gemini-10X

Na rynku angielskim **ukazały się pamięci typu EPROM produkcji Texas Instrument. Pamięć TMS27C256 wykonana jest w techno-logii HVCMOS, co zapewnia dużą szybkość działania i współpracę z układami MOS i bipolarnymi. Organizacja pamięci — 32k x 8 bitów. Wyprowadzenia kompatybilne z istniejącymi pamięciami EPROM 64k i 128k. Pamięć pracuje przy zasilaniu pojedynczym napięciem +5V. Wszystkie wejścia i wyjścia zapewniają standardowe poziomy TTL. Wyjścia danych są trójstanowe, umożliwiając łączenie bloków pamięci do wspólnej szyny danych.**

Szwedzka firma Coiltronics HB **opracowała – tester linii komunikacyjnych systemu RS232/V24. Tester o nazwie Comtest kontroluje trójstanowo jednocześnie 23 sygnały przy pomocy wbudowanych diod elektroluminescencyjnych typu LED. Może również sprawdzać pętle prądowe dwukierunkowych linii kontrolnych dla prądów 10, 20,40 i 60 mA. Przy użyciu testera można przerwać lub zmienić kierunek każdej z 25 linii posługując się wbudowanymi przełącznikami i zworami dostarczanymi razem z urządzeniem.**

Generator zapachu. **Zachodnioniemiecka firma Odorant Elektronik GMBH wyprodukowała pod nazwą CD4711 czterobitowy programowalny generator ... zapachu z myślą o współpracy tego bioelektronicznego układu z cyfrowymi odbiornikami telewizyjnymi. Układ może wytworzyć do 15 różnych zapachów wybieranych stanami wejść danych. Producent zapowiada również wersję ośmiobitową z modulacją intensywności zapachu.**

British Software Factory **opracowała procesor do bezpośredniego przetwarzania tekstów przeznaczony do pracy w systemach „poczty elektronicznej". Kość ta nazwana „krzemową pocztą" pozwala użytkownikom na bezpośrednie wysyłanie lub odbiór informacji do (lub z) obszaru przetwarzania tekstowego. Umożliwia ona skracanie i łączenie, przesuwanie w poziomie, wyszukiwanie i zamianę, wyświetlanie ostatecznej formy, drukowanie podczas edytowania, sortowanie i liczenie, wejście wielokolumnowe, jednocześnie przetwarzanie dwóch dokumentów i wiele innych operacji dokonywanych na przesyłanym materiale. Posiada również układ szyfrujący operujący na zwykłych zbiorach typu ASCII, umożliwiający utajnianie pożądanych informacji.**

Sircal Instruments **wprowadziła do sprzedaży nowy system typu „cartridge" umożliwiający tworzenie własnych programów w BASIC-u lub języku maszynowym. System bazuje na 8k pamięciach typu EPROM i może być używany zarówno przez amatorów jak i profesjonalistów wykorzystujących minikomputery Commodore 64 i 128. Zaprogramowana jednorazowo pamięć umożliwia natychmiastowe wykorzystywanie zawartego w niej programu przez przyłączenie „cartridge'a" do komputera za pośrednictwem systemu o nazwie EPILOG-1.**

Dla inżynierów i konstruktorów **systemów mikroprocesorowych firma Thorn Emi Instruments opracowała zestaw pomiarowy współpracujący z większością komputerów osobistych posiadających system MS-DOS lub CP/M. Utworzony w ten sposób system ułatwia uruchamianie opracowywanych systemów mikroprocesorowych. Zestaw o nazwie UDL (Universal Development Laboratory) czyli Uniwersalne Laboratorium Badawcze zawiera 48-kana-łowy kontroler stanu szyn systemowych, uniwersalny 8/16 bitowy emulator, programator pamięci EPROM i programowalny generator. UDL umożliwia wyszukiwanie błędów sprzętowych i programowych w systemach pracujących z 36 typami mikroprocesorów.**

C128

Zaczęło sie w styczniu 1985 w Las Vegas (USA). Wtedy to firma Commodore zaprezentowała 3 nowe mikrokomputery: Commodore PC-128. Commodore PC-128D i Commodore LCO. Commodore LCO jak dotąd nie pokazał się w sklepach, natomiast podróż PC-128 z USA do Europy trwała przeszło pół roku. Na początku lipca 1985 trafił na półki sklepowe i do katalogów domów wysyłkowych, ściślej biorąc miał trafić, bo w sklepach pojawiły się tylko pojedyńcze egzemplarze. Firma milczała, handlowcy i domy wysyłkowe też, a chętni na PC-128 ostrzyli sobie zęby i ... czekali. Krążyły różne plotki: Commodore zbankrutował, cała partia komputerów jest uszkodzona itd. Milczenie i niepewność trwały do początku października 1985. Firma wyjaśniła, że przyczyną opóźnienia w dostawie był błąd w pamięci ROM.

Początki nie były więc najlepsze, ale przyszłość obiecująca. PC-128D ukazał się dopiero w połowie stycznia 1986.

Znany slogan reklamowy: „PC-128 to trzy komputery w jednym" jest całkowicie słuszny, bo PC-128 może pracować w jednym z trzech trybów: **C-64, C-128 i CP/M**

OD ŚRODKA

Spróbujmy zdjąć pokrywę i zajrzeć do wnętrza komputera. Nic ciekawego nie zobaczymy, bo całą płytkę zasłania radiator, którego elementy są przyklejone do układów scalonych za pomocą specjalnej pasty silikonowej, która przewodzi ciepło. Ale po zdjęciu radiatora ... Pierwsze, co rzuca się w oczy to 2 mikroprocesory: **8502** — kompatybilne z **6502/6510 i Z80**. Zauważymy wszystkie układy, które występują w **C-64**, a więc: **SID** (układ dźwiękowy — **S**ound **I**ntertace **D**evice), **VIC** (procesor graficzny — **V**ideo **I**nterface **C**hip) i dwa porty we/wy **CIA**) **C**omplex **I**nterface **A**dapter). A z nowości: 128K RAM, 2K CMOS RAM, układ zarządzania pamięcią **MMU** (**M**emory **M**anagement **U**nit), drugi procesor graficzny **VDC**, układ logiczny **PLA** i 64K pamięci **ROM** (4 kości po 16K). Jest także miejsce na 32K dodatkowei pamięci ROM.

KLAWIATURA

Tyle w środku, a na zewnątrz? Doskonała profesjonalna klawiatura z wyodrębnionym blokiem dziesiętnym i 16-toma klawiszami funkcyjnymi umieszczonymi w jednym rzędzie. Czterem z klawiszy funkcyjnych można przyporządkowac 8 dowolnych funkcji. Funkcje pozostałych zostały określone przez konstruktorów. Cztery z nich służą do sterowania kursorem, a dalsze 8 to: **ESC, TAB, ALT, ASCII/DIN, HELP, LINE FEED, 40/80 DISPLAY i NO SCROLL**. Ich działanie będzie omówione w dalszej części artykułu. Warto podkreślić, ze choć nie jest to klawiatura kontaktronowa, w niczym nie ustępuje klawiaturom w pełni profesjonalnym.

Wersja niemiecka PC-128 posiada dwa zestawy znaków, a właściwie dwa różne układy klawiatury. Pierwszy z nich jest opracowany wg standardu **ASCII**, a drugi wg **DIN**. Znaki ASCII są naniesione na kla wisze czarną farbą, natomiast zestaw DIN jest namalowany kolorem szarym. Do przełączania służy klawisz ASC/DIN. W zestawie DIN są dostępne niemieckie znaki (ä, ö, ß), a układ klawiatury jest taki, jak w maszynie do pisania. Jako ciekawostkę podam, że znaki DIN można uzyskać także w trybie C-64, jednak tylko przez kody CHR$. PC-128D nie różni się działaniem od PC-128. Ma tylko w jednej obudowie komputer i stację dysków oraz osobną klawiaturę (podobnie jak IBM czy Amiga).

KOMPATYBILNOŚĆ Z C-64

Po włączeniu komputera, zgłasza się on komunikatem, który informuje o ilości wolnej pamięci i producencie. Komputer pracuje w trybie **PC-128**. Łatwo można przejść do trybu C-64. Wystarczy napisać **GO 64** i nacisnąć **RETURN**. Ponieważ program będący aktualnie w pamięci zostanie skasowany, komputer upewnia się, pytając: **ARE YOU SURE?** na co trzeba odpowiedzieć **Y** lub **YES**. Jest też inny sposób: podczas włączania trzymać wciśnięty klawisz **COMMODORE**

Po przejściu do trybu C-64 komputer zgłasza się identycznym napisem jak „normalny" C-64. Taki też mamy do dyspozycji. Producent zapewnia o 100% kompatybilności z oprogramowaniem i osprzętem C-64. W rzeczywistości sprawdza się to w 99,9%. Nieprawdą są pogłoski mówiące, że bardzo dużo programów napisanych na C--64 nie działa. Jak dotąd znane mi są dwa takie programy: ELITE I EMPIRE (gry firmy FIREBIRD). Jest ich na pewno więcej, ale przypuszczam, że są to ilości nie przekraczające kilkunastu pozycji. Istotną wadą jest brak możliwości korzystania w trybie C-64 z klawiatury dziesiętnej i dodatkowych klawiszy funkcyjnych (z wyjątkiem F1-F8). Programiści znaleźli sposób i na to. Powstał krótki program, który umożliwia wykorzystanie klawiatury dziesiętnej w trybie C-64

STACJA DYSKIETEK

Jeżeli do komputera podłączona jest stacja dysków, to zaraz po jego włączeniu zielona dioda (Uwaga! o nowych stacjach zielona dioda oznacza zapis/odczyt a czerwona zasilanie) zacznie migać, a głowica stukać. Jeśli w stacji nie ma dyskietki, to po chwili ukaże się napis READY i kursor. W ten sposób komputer realizuje funkcję „**auto-boot**". Robi to zawsze po włączeniu po reset. Procedura ta ładuje program, który znajduje się na pierwszej ścieżce dyskietki. Jest to dużym ułatwieniem, gdyż oryginalne programy ładują się same.

Skoro w stacji nie było dyskietki — komputer oczekuje na nasze polecenia.

CO NOWEGO

Zapoznajmy się z możliwościami, jakie oferuje nam PC-128. Najpierw zdecydujmy, czy chcemy pracować w trybie 40 czy 80 znaków. Wyboru dokonuje się klawiszem **40/80 DISPLAY**. Niektóre programy same dostosowują się do wybranego trybu.

Jeżeli już wybraliśmy tryb pracy, poznajmy możliwości edytora ekranowego. Dostęp kursorem mamy do całego ekranu. W dowolnym jego miejscu można zacząć pisać tekst programu w BASIC-u. Nowością jest klawisz **ESC**. Po jego wciśnięciu razem z którymś znakiem włącza się jedna z wielu dodatkowych funkcji edytora ekranowego. Np. wymazanie ekranu od/do kursora, wymazanie linii, w której znajduje się kursor lub jej części od/do kursora, przesunięcie całej linii do góry, w dół i wiele innych. Mocną stroną PC-128 jest BASIC. Do dyspozycji programisty stoi 165 rozkazów i funkcji. Tak często używany w C-64 **POKE** zszedł na pozycję drugoplanową. Na program zarezerwowanych jest ok. 122 KB pamięci, z tym, że jedna połowa na tekst programu (**BANK 0)**, a druga połowa na zmienne i tablice (**BANK 1)**. Komputer podczas pracy sam przełącza banki pamięci. W tym miejscu należy wyjaśnić, że pamięć RAM jest podzielona na dwa banki po 64K. Przełączanie dokonywane jest automatycznie lub przez programistę (instrukcja **BANK**). W **BASIC--u 7.0** można wyróżnić, oprócz podstawowego zestawu instrukcji, dodatkowe komendy przeznaczone do sterowania grafiką, sprite'ami, dźwiękiem, stacją dysków.

DANE TECHNICZNE:

MIKROPROCESOR: 8502 (kompatybilny z 6502); Z80A

SYSTEM PRACY: Basic 7.0; Basic 2.0; CP/M 3.0

POJEMNOŚĆ PAMIĘCI: RAM 128KB, ROM 64KB

ILOŚĆ ZNAKÓW W LINII: 40 lub 80

KOLOR: 16 barw

DŹWIĘK: 6581 SID Chip

PAMIĘĆ MASOWA: magnetofon kasetowy, stacja dyskietek elastycznych

KLAWIATURA: maszynowa z wyodrębnionym blokiem cyfrowym



COMMODORE 128

GRAFIKA

Możliwości graficzne PC-128 są bardzo duże. Można pracować w jednym z 6 trybów graficznych z rozdzielczością 160x200, 320x200 lub 640x200. W trybie 160x200 można jednocześnie wykorzystać 4 kolory z palety 16, a w trybach 320x200 i 840x200 — 2 kolory. Pamięć obrazu graficznego zajmuje 9K i jest rezerwowana z obszaru dostępnego dla BASIC-a.

Istniejące rozkazy graficzne umożliwiają narysowanie praktycznie każdej płaskiej figury geometrycznej. Interesująca jest możliwość tworzenia tzw. **SHAPE**. Są to wycinki obrazu graficznego zapamiętane pod zmienną alfanumeryczną. Można je umieszczać w dowolnym miejscu obrazu graficznego w pozytywie, negatywie lub nakładać na już istniejący rysunek.

Posiadacze C-64 wiedzą, jak mozolnie projektowało się na nim **SPRITE'y**. W PC--128 sprawę rozwiązano o wiele lepiej. Jednym z rozkazów BASIC-a włącza się **sprite-editor** (podobny do Animy na ZX Spectrum). Zaprojektowanie sprite'a nie przedstawia już trudności. Wystarczy poruszać się kursorem w polu o wymiarach **sprite'a** (jednemu elementowi sprite'a odpowiada kratka 8x8 punktów) i w odpowiednim miejscu ustawiać punkty. Gotowy **sprite** można zapamiętać pod zmienną alfanumeryczną. Poruszanie po ekranie zrealizowane jest na przerwaniach, w związku z czym **sprite'y** są zupełnie niezależne od innych czynników. Podajemy kąt i szybkość poruszania – z **sprite** porusza się.

Niestety rozkazy graficzne BASIC-a działa tylko w trybach 160x200 i 320x200. Tryb 640x200 jest dostępny tylko dla programujących w assemblerze i tylko na scecjalnym monitorze dołączonym przez wyjście **RGB**. Programiści są niezawodni. Rozwiązali i ten problem. Powstało rozszerzenie, które umożliwia wykorzystanie grafiki 640x200 za pomocą opisanych rozkazów graficznych.

Dzięki temu, że PC-128 ma dwa niezależne procesory graficzne (**VIC** i **VDC**) — może generować jednocześnie dwa różne obrazy. Jeden na monitorze 40-znakowym, a drugi na 80-znakowym. Jako monitor 80-znakowy można podłączyć (kwestia odpowiedniego kabla) dowolny monitor z wejściem Video, np. Neptun. W obu trybach graficznych można zdefiniować 1 okienko **WINDOW**).

MUZYKA

Przewidziano także coś dla tych, którzy interesują się muzyką. Dźwięk można uzy-skać dwoma sposobami. Pierwszy polega na dobraniu wszystkich parametrów (obwiednia, wysokość, filtry, wzmocnienie) tak aby uzyskać żądane brzmienie. Drugi jest prostszy. Podajemy tylko, na jakim instrumencie komputer ma grać (wpisujemy komputerowe oznaczenia nut). PC-128 ma trzy generatory dźwięku i jeden generator szumów. Dźwięk realizowany pierwszym spo sobem wykorzystuje przerwania.

BASIC 7.0

Oprócz rozkazów wybitnie specjalistycznych, są też podstawowe. **BASIC 7.0** jest trochę zbliżony do Pascala. Bardzo ułatwia programowanie strukturalne. Do tego celu służy komenda **ELSE**, pętla warunkowa **DO LOOP UNTIL/WHILE** i zestaw **BEGIN/BEND**. Szkoda tylko, że nie ma możliwości definiowania procedur (jak np. w **Simon's BASIC-u** na C-64). Jeżeli popełnimy błąd — można go łatwo odszukać przez naciśnięcie klawisza **HELP**.

Dla programujących w assemblerze jest MONITOR dostępny bezpośrednio z BASIC-a. Jego wywołanie nie powoduje skasowania programu aktualnie znajdującego się w pamięci, a powrót do BASIC-a jest możliwy w dowolnym momencie. Zestaw komend monitora pozwala na pisanie nawet skomplikowanych procedur w języku wewnętrznym.

CP/M

Trzecim trybem pracy PC-128 jest CP/M. System trzeba załadować z dyskietki dołączanej przez producenta. CP/M wykorzystuje procesor **Z80** z zegarem 4MHz. **CP/M** umożliwia wykorzystanie ogromnej biblioteki programów napisanych na niego. Na PC--128 pracuje **CP/M +** , wersja 3.0. Oferuje on szereg rozkazów umożliwiających tworzenie własnych plików, dokonywanie operacji na nich, kopiowanie plików i inne.

PERYFERIA

Dp PC-128 producent oferuje kilka nowych urządzeń peryferyjnych. Są dwa nowe kolorowe monitory o wysokiej rozdzielczości: 1901 i 1902, które umożliwiają odbiór obrazu o bardzo wysokiej jakości, zarówno w trybie 40, jak i 80-znakowym. Jest też oryginalna mysz Commodore 1530.

Dłuższego omówienia wymagają trzy nowe stacje dysków: **1570, 1571 i 1572**.

Stacja **1570** to nowy hardware w starej obudowie od stacji **1541**. **1570** posiada jeden napęd z możliwością odczytu jednostronnego. Od 1541 różni się tylko szybkością zapisu (odczytu, która wynosi dot, także pozostałych dwóch stacji): w trybie C-64 : 3800 bodów w trybie PC-128 : 15000 bodów w trybie CP/M : 35000 bodów **1571** ma w sobie prawie drugi komputer Ma wbudowany własny mikroprocesor 6502, 32K ROM (zawiera DOS — Disk Operating System), 2K bufor RAM i 2 porty we/wy 6522 i



COMMODORE 128D

6526. 1571 posiada dwie głowice, można więc zapisywać dyskietki dwustronnie. Jest (jak zapewnia producent) całkowicie kompatybilna z 1541. Z tą kompatybilnością bywa różnie Jeżeli komputer pracuje w trybie 64, stacja przełącza się na tryb pracy stacji 1541 i wytacza jedną głowicę. Okazuje się, że producent trochę zmienił ROM nowej stacji i w rezultacie tej przeróbki niektóre programy kopiujące (Turbocopy, Quickcopy i Fastcopy) przeznaczone dla C-64 z 1541 nie chcą działać. Problem ten dotyczy tylko kilku programów kopiujących, pozostałe działają normalnie). Wreszcie trzecia stacja — **1572**. Ten model ma wbudowane 2 napędy **1571** i są to właściwie 2 stacje 1571 w jednej obudowie.

Do PC-128 można podłączyć w sumie 8 napędów, z tym, że mogą być max. 4 stacje (adresy 8-11) po 2 napędy (0 i 1 lub A i B w CP/M.)

Do współpracy z dyskami opracowano sporo komend. Z ich pomocą można nagrywać, ładować i sprawdzać programy, tworzyć i uzupełniać pliki danych, formatować dyskietki i wreszcie kopiować poszczególne programy (pola) lub bezpośrednio całą dyskietkę. Niestety, rozkazy kopiujące (**COPY** i **BACKUP**) działają tylko z podwójnym napędem 1572, a nie można wykorzystać dwóch stacji 1570 lub 1571.

Nowe stacje mogą leż pracować ze zwykłym **C-64, a 1541** z **PC-128**. Pojemność jednej dyskietki (format 1571 — czyli obie strony) wynosi w trybie PC-128 — 346K, a w CP/M — 410K.

W wersji podstawowej PC-128 ma 128 K RAM. Można dokupić moduł rozszerzający pamięć RAM do 512K Dodatkowe 384K mogą być wykorzystane jako RAM-Disk. Do tego celu służą trzy rozkazy BASIC-a: **FETCH, STASH i SWAP.**

Na początku wspominałem o wolnym miejscu na 32k ROM. Zostało ono już wykorzystane. Firma Sas-Bernd oferuje za 298 DM pamięć ROM z rozszerzeniem BASIC-a. Moduł nazywa się **HIGHWAY** i daje ponad 200 nowych komend i wiele innych możliwości.

OPROGRAMOWANIE

Wg specjalistów C-64 jest najlepiej opro gramowanym komputerem domowym. PC-128 stał się więc drugim najlepiej oprogra

mowanym komputerem domowym. Istnieje już także sporo programów pracujących w trybie 128.

Warto wymienić kilka tytułów: **Textomat Plus 128, Jane 128, Last V8, Superbase Plus 128, Jane 128, Last V8, Superbase 128, Financial Planner, Swift Calc Data Manager** i kilkadziesiąt (na razie) innych.

Dużo programów mogą wykorzystać użytkownicy CP/M. Np. **Turbo-Pascal** działający pod CP/M PC-128 jest identyczny z wersją na IBM PC.

Pod CP/M działają także inne języki: **Cobol, Fortran, Basic, Prolog. Lisp, Ada** Opracowano także wersje takich gwiazd, jak: **dBase II, Wordstar i Multiplan**

LITERATURA

Znana firma DATA BECKER wydała już przeszło 10 książek o PC-128. Oto tytuły kilku ciekawszych: **Das Grosse Grafik Buch fuer PC-128, Tips Tricks fuer PC--128. Peek's und Poke's fuer PC-128 i Das Grosse CP/M Buch.**

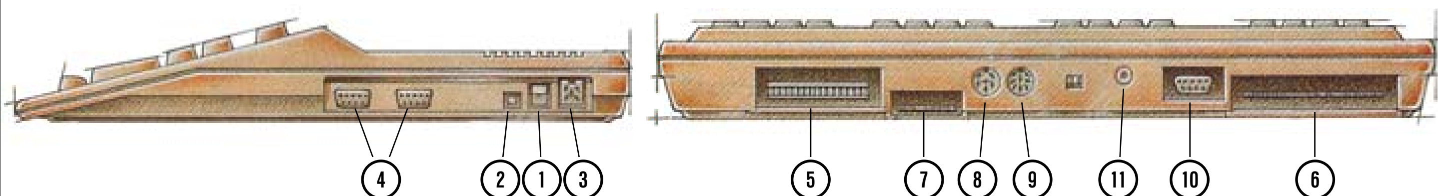
Na zakończenie aktualne ceny (lipiec):

PC-128	800 DM
PC-1280	1600 DM
1901	900 DM
1902	1300 DM
1570	600 DM
1571	1000 DM
1530	150 DM

PC-128 pragnąłbym polecić wszystkim tym, którzy chcą robić coś więcej, niż tylko grać. Nadaje się doskonale do obliczeń, obróbki tekstu i danych, wszelkich prac inżynierskich.

Przemysław Koziarski

Gniazda Wejścia/Wyjścia



1. *zasilanie* **2**. *reset* **3**. *wyłącznik* **4.** *drążki sterowe, wioselka (paddle) i pióro świetlne* **5.** *rozszerzanie pamięci, podłączanie modułów oraz dysku twardego* **6.** *we/wy (w standardzie RS 232) modem, interface do drukarki i stacji dyskietek* **7.** *magnetofon* **8.** *drukarka, ploter, stacja dyskietek* **9.** *wyjście monitorowe (40 znaków)* **10.** *wyjście monitorowe (80 znaków)*



PORADNIK MŁODEGO PIRATA cz. II

Czy wiesz drogi Czytelniku w jaki sposób Twój Commodore rozpoznaje daną konfiguracje pamieci?

Służą mu do tego celu komórki o adresach **43, 44, 45, 46, 55, 56 ($, 2B, 2C, 2D, 2E, 37 i 38)**. Tak naprawdę to system operacyjny zbiera dane ze znacznie większej ich ilości, ale wszystkie powyżej wymienione mają kluczowe znaczenie przy rozpoznawaniu konfiguracji. Poniżej zamieszczamy ich skrótowy opis:

43—44 ($2B—2C) — określają początek pamięci RAM przeznaczonej dla użytkownika. Od adresu (zawartego w tych komórkach) będzie przechowywany Twój program.

45—46 ($2D—2E) — określają adres pierwszej komórki pamięci po zakończeniu Twojego programu (nie mylić z końcem pamięci!) oraz początek obszaru pamięci przeznaczonego dla zmiennych.

55—56 ($37—38) — zawierają najwyższy nieprzekraczalny adres pamięci, w której działa interpreter języka BASIC.

Możliwość regulacji obszaru pamięci, w której działa BASIC jest dla nas bardzo przydatna w wielu przypadkach. Wyobraźmy sobie bardzo długi program, zawierający np. wielką ilość danych np. dla sprite'ów. Wykorzystaliśmy już cały dodatkowy obszar pamięci od adresu **49152 do 53247 ($C000 — CFFF)** i potrzebujemy jeszcze 300 bajtów, ale z gwarancją niezakłócania tego obszaru przez np. BASIC. Wystarczy wtedy tylko obniżyć górną granicę o rzeczone 300 bajtów.

Liczby zawarte w opisywanych komórkach zapisane są w standardowej formie — niższy bajt i wyższy bajt. Odczytać je można za pomocą funkcji PEEK (X). Jak jednak wylicza się z nich adres? Bardzo prosto. W tym celu należy pomnożyć wartość wyższego bajtu przez 256 i do otrzymanego wyniku dodać wartość niższego bajtu.

PRINT PEEK (46) 256 + PEEK (45)
2051
READY.

Komórki o adresach **43, 45 i 55** zawierają zawsze młodszy bajt zaś **44, 46 i 56** starszy. Dla wprawy możesz spróbować wpisać jakiś krótki program i sprawdzić do której komórki w pamięci on sięga. Wystarczy jeżeli po jego wpisaniu napiszesz:

PRINT PEEK (43) + PEEK (44) * 256 : REM POCZATEK
PRINT PEEK (45) + PEEK (46) * 256 : REM KONIEC PROGRAMU + 1 PRINT PEEK (55) + PEEK (56) * 256 : REM KONIEC PAMIECI

Jeżeli chcesz poeksperymentować to usuń teraz z programu którąś linię i wykonaj powyższą operację ponownie; zobaczysz wtedy jak obszar pamięci zajmowany przez Twój program zmniejsza się.

Czy to już wszystko? Oczywiście, że nie. Nie wolno nam przecież zapominać o zmiennych, których wartości komputer musi także gdzieś przechowywać. Wydzielonym do tego miejscem jest obszar pamięci zaczynający się bezpośrednio po końcu Twojego programu + 1. Innymi słowy, jeżeli obliczony z komórek adres wynosi np. 10167 oznacza to, że od tego adresu zaczyna się Twój obszar dla zmiennych, podczas gdy program kończy się w komórce 10166. Podczas wprowadzania poprawek (np. usuwanie czy wpisywanie linii programu) system operacyjny w ogóle się tym obszarem nie zajmuje; rozkaz RUN jest zawsze poprzedzony wykonaniem procedury CLR kasującej wartości wszystkich zadeklarowanych zmiennych.

Sprawa byłaby prosta, gdyby takie ułożenie obszaru pamięci dla zmiennych dotyczyło wszystkich ich rodzajów. Niestety (ku utrapieniu!) o ile zmienne numeryczne są przechowywane właśnie w ten sposób — czyli zaraz po końcu programu — o tyle zmienne łańcuchowe (np. OK$ = „BAJTEK") przechowywane są w sposób zgoła odmienny. Wydzielony dla nich obszar zaczyna się zawsze od adresu określonego w komórkach 55 i 56 i zbiega w DÓŁ w miarę deklarowania następnych zmiennych tego typu: Na pewno podczas przegrywania programów na inną kasetę zdarzył Ci się kiedyś taki zabawny przypadek: napisałeś SAVE „nazwa" bądź ←S „nazwa" i... kom-

```
0   2049                     40960 65535
|    | --->   |-->|-->|-->|<--|      |
|    |   A    | B | C | D | E |      |
$0  $0800                    $A000 $FFFF
```

Rys. 1
A – obszar pamięci zajęty przez program
B – obszar pamięci zmiennych numerycznych
C – obszar pamięci dla tablic (A(X), BG(Y))
D – obszar pamięci wolnej
E – obszar pamięci dla zmiennych łańcuchowych
Strzałki oznaczają kierunki zajmowania pamięci w miarę deklarowania nowych zmiennych.

```
10 PRINT CHR$(147)
20 INPUT"NAZWA ZBIORU  :  ";F$
30 OPEN 1,8,3,"0:"+F$
40 GET#1,L$:GET#1,H$
50 X$=H$:GOSUB 100:H=X*256
60 X$=L$:GOSUB 100:L=X
70 IF H+L=51143 THEN PRINT "NIE MA
   TAKIEGO
   ZBIORU !":CLOSE 1:RUN
80 PRINT "ADRES =  ";H+L: CLOSE 1: END
90 IF X$="" THEN X=0: RETURN
99 X=ASC(X$): RETURN
```

puter nagle zasygnalizował Ci błąd braku pamięci!

?OUT OF MEMORY ERROR
READY.

Jeżeli byłeś uparty to już po paru próbach odkryłeś sposób na Twój krnąbrny program — wystarczyło go zapisać bez tytułu. Tym razem odpowiedź komputera brzmiała prawidłowo:

PRESS RECORD AND PLAY ON TAPE

Co się właściwie stało?

Otóż tytuł każdego programu jest traktowany jako specyficzna zmienna łańcuchowa, która jest przechowywana w obszarze pamięci przeznaczonym dla tego typu zmiennych. Błąd braku wolnej pamięci sygnalizowany był prawidłowo, gdyż widocznie przepisywany program nie pozo-



Rys. 2
A – obszar pamięci zajmowany przez program
B – obszar pamięci pozostawiony dla zmiennych łańcuchowych C – obszar konieczny do wpisania tytułu.

stawił już miejsca w pamięci na wpisanie jego tytułu. Innymi słowy konfiguracja pamięci wyglądała następująco:

Czasami wystarczy tylko zredukowanie ilości znaków w tytule, czasami zaś program nie da się przepisać w żaden sposób. Warto także zaznaczyć, że wpisanie programu bez tytułu możliwe jest jedynie na taśmie, gdyż system operacyjny dysku w żaden sposób nie pozwoli na wpisanie na dyskietkę programu pozbawionego tytułu. Spróbujmy doświadczalnie zbadać zapis z tytułem i bez niego. W tym celu musimy się posłużyć małym oszustwem, gdyż wpisywanie tak długiego programu byłoby nonsensem; ponieważ już wiesz, które komórki decydują o końcu programu, możesz ich użyć do podstępnego okłamywania komputera:

POKE 43,1 : POKE 44,8 : REM USTAWIANIE POCZĄTKU PAMIĘCI
POKE 44,255 : POKE 46,169 : REM ADRES KOŃCA PROGRAMU – 40959 CLR : REM NIEZBĘDNE DO USTAWIENIA POCZĄTKÓW OBSZARÓW DLA ZMIENNYCH

Teraz — po upewnieniu się, że kaseta w magnetofonie może być skasowana — spróbuj zapisać swój „program” na taśmie:

SAVE "TEST 1" (lub ←S "TEST 1")
?OUT OF MEMORY ERROR
READY.

Widzisz? Nie ma już miejsca w pamięci, aby Twój „tytuł” zapisać. Spróbujmy więc go odrzucić:

SAVE (lub ←S)
PRESS RECORD AND PLAY ON TAPE

Tym razem komputer nie sygnalizuje już błędu ...

(kd)

PROSTE ZABEZPIECZENIE DLA C 64

<u>Nie ma i nie będzie nigdy tak zabezpieczonego programu, którego nie można by odbezpieczyć: zawsze będzie to tylko kwestia czasu. Dla tych wszystkich, którzy chcą układane przez siebie programy zabezpieczyć w jakikolwiek sposób przed niepowołaną ingerencja mamy opisane poniżej, proste rozwiązanie.</u>

Zabezpieczenia przed wylistowaniem programu mają już co najmniej tak samo długą historię jak programy komputerowe. Techniki te są coraz bardziej udoskonalane, ale nie łudźmy się — dla eksperta od danego typu komputera złamanie blokady zabezpieczającej to tylko kwestia czasu. Ponadto nie zawsze jest to konieczne, gdyż istnieją już programy umożliwiające „podglądanie” zawartości pamięci czy dyskietki, nie mówiąc już o taśmie. Programy komputerowe są zresztą rzadko kiedy zabezpieczane przed podglądaniem — rzeczywistości chodzi tu raczej o uniemożliwienie ich kopiowania, które stanie się trudniejsze jeśli nie wiadomo co siedzi w programie.

Opisany poniżej sposób jest bardzo prosty i w zasadzie nawet średnio zaawansowany programista szybko taki program odbezpieczy. Tym niemniej wydaje się nam jednak, że warto go podać.

Gdy program, który ułożyłeś uznasz za gotowy do zapisania na taśmie czy dysku wykonaj najpierw następująca operację:

PRINT PEEK (2050)

Po otrzymaniu wyniku zanotuj go i zabezpiecz program za pomocą:

POKE 2050, LICZBA gdzie LICZBA jest w zakresie 0-255, Wpisanie w tę komórkę wartości 0 spowoduje całkowite zablokowanie listowania; przy innych liczbach część programu może się jednak na ekranie ukazać. Tak zabezpieczony program zapisz teraz na taśmie czy dysku.

Po ponownym wczytaniu programu do pamięci, przed jego uruchomieniem musisz wpisać ponownie w komórkę 2050 wykreśloną uprzednio wartość. Bez jej wpisania programu nie można uruchomić ani też wylistować (w normalny sposób — za pomocą LIST). W niektórych jedn ak wypadkach (np. przy stosowaniu jednocześnie pewnych programów maszynowych) blokada ta może okazać się nieskuteczna.

(kd)

COŚ DLA KOLEKCJONERÓW

Do napisania tego programu skłoniła nas potrzeba chwili. Przy porządkowaniu stosu dyskietek przekonaliśmy się na własnej skórze jak niewygodny w użyciu jest BASIC V2.0 firmy Commodore. Od ciągłego wypisywania LOAD „$”, 8 mogą porządnie zaboleć palce nie mówiąc już o zużywaniu klawiatury. W związku z tym proponujemy przy wszelkich pracach porządkowych przeprowadzonych na dyskietkach zastosować poniżej opisany program. Umożliwia on wczytywanie katalogu dyskietki do pamięci komputera jedynie poprzez wciśnięcie klawisza F7; wylistować zaś go można używając F1.

Program ten wykorzystuje wewnętrzny system przerwań komputera w związku z czym w niektórych wersjach C-64 mogą wystąpić pewne anomalia związane z ruchem kursora; jest on znacznie szybszy niż normalnie.

Na marginesie systemu przerwań warto wspomnieć, że zastosowaliśmy tu nieco zmodyfikowany sposób włączenia tej procedury w przerwania, umożliwiający bezpieczne stosowanie programu na wszystkich wersjach Commodore 64. Zwracamy na to uwagę celowo, gdyż w międzyczasie dotarło do nas parę uwag, że wcześniej opisywany przez nas program „BEEP 64” („Bajtek” 3-4) nie chciał działać na niektórych wersjach Commodore 64.

Klaudiusz Dybowski
Michał Silski

```
100 REM ******************************
101 REM *                            *
102 REM *   C-64  QUICK  DIRECTORY   *
103 REM *                            *
104 REM *   MANIAK  COMPUTER  CLUB   *
105 REM *                            *
106 REM *   W-WA,  WASILKOWSKIEOO 7  *
107 REM *                            *
108 REM ******************************
109 :
110 PRINT CHR$(147):A$=CHR$(34):PRINT"   F1 = LIST";
115 PRINT TAB(22)"F7 = LOAD ";A$;"$";A$;" ,8"
120 FOR J = 49152 TO 49247:READ Q:CK = CK+Q:POKE
J,Q:NEXT
130 IF CK<>11342 THEN PRINT "ERROR IN DATA !":END
140 SYS 49152:NEW
145 :
150 DATA 120,173,020,003,141,065,192,173,021,003,141
168 DATA 066,192,169,025,141,020,003,169,192,141,021
170 DATA 003,088,096,032,159,255,166,198,240,032,202
180 DATA 189,119,002,201,133,208,003,076,050,192,201
190 DATA 136,208,017,076,067,192,160,255,200,232,185
200 DATA 083,192,157,119,002,208,246,134,198,076,000
210 DATA 000,160,255,200,232,185,087,192,157,119,002
220 DATA 208,246,076,062,192,000,076,201,013,000,076
230 DATA 207,034,036,034,044,056,013,000
```

Drogi Czytelniku, „Bajtek” daje Ci w tym numerze niepowtarzalną szansę; nie wychodząc z domu i nie mając grosza przy duszy zagrasz z największym na świecie połykaczem pieniędzy zwanym, „jednorękim bandytą”. Bywalcy kasyn twierdzą, że pociąganie za „rękę” tego automatu kończy się zazwyczaj zniknięciem zawartości Twojego portfela. A Ty nie stracisz ani centa (fakt, że również nic nie wygrasz pomińmy milczeniem; najważniejsza jest przecież sama gra). Wystarczy trochę dokładności i cierpliwości przy wprowadzaniu zamieszczonego obok programu. Ponieważ zakładamy, że jako porządny obywatel nie bywasz w kasynach, opiszemy Ci jak spróbować ograć „jednorękiego”.

```
 1 '
 10 '************************
 20 '*  JEDNOREKI BANDYTA   *
 30 '*                      *
 40 '*autor:O.Felchner/RFN *
 50 '*polska wersja: S.W.   *
 60 '************************
 70 '*  Definicja  znakow   *
 80 SYMBOL AFTER 200
 90 SYMBOL 224, 3, 7, 7, 15,
   15, 15, 15, 15
 100 SYMBOL 225, 255, 249, 253,
     252, 254, 254, 254, 254
 110 SYMBOL 226, 192, 224, 224,
     240, 240, 240, 240, 240
 120 SYMBOL 227, 15, 15, 31, 31,
     63, 127, 255, 0
 130 SYMBOL 228, 254, 254, 255,
     255, 255, 255, 255, 60
 140 SYMBOL 229, 240, 240, 120,
     56, 188, 158, 255, 0
 150 SYMBOL 231, 0, 0, 0, 0, 61,
     126, 255, 255
 160 SYMBOL 232, 0, 0, 56, 68,
     130, 2, 2, 2
 170 SYMBOL 233, 1, 1, 0, 0, 0,
     0, 0, 0
 180 SYMBOL 234, 249, 249, 254,
     124, 56, 0, 0, 0
 190 SYMBOL 235, 2, 5, 1, 0, 0,
     0, 0, 0
 200 SYMBOL 230, 0, 0, 0, 0, 0,
     0, 0, 0
 210 SYMBOL 200, 224, 224, 64,
     112, 56, 60, 30, 31
 220 SYMBOL 201, 28, 28, 8, 24,
     60, 60, 126, 255
 230 SYMBOL 202, 14, 14, 4, 12,
     28, 32, 72, 208
 240 SYMBOL 203, 15, 15, 15, 15,
     0, 0, 0, 0
 250 SYMBOL 204, 255, 255, 255,
     255, 0, 0, 0, 0
 260 SYMBOL 205, 160, 96, 96,
     96, 0, 0, 0, 0
 270 SYMBOL 206, 0, 127, 224,
     239, 224, 224, 224, 224
 280 SYMBOL 207, 0, 255, 0, 242,
     18, 34, 66, 130
 290 SYMBOL 208, 0, 254, 2, 242,
     154, 138, 154, 242
 300 SYMBOL 209, 225, 226, 228,
     232, 239, 224, 255, 255
 310 SYMBOL 210, 2, 2, 2, 2,
     242, 0, 255, 255
 320 SYMBOL 211, 194, 194, 194,
     194, 194, 2, 254, 252
 330 SYMBOL 212, 0, 0, 96, 112,
     120, 124, 125, 125
 340 SYMBOL 213, 0, 0, 0, 0, 0,
     0, 255, 255
 350 SYMBOL 214, 0, 0, 0, 3, 6,
     118, 254, 255
 360 SYMBOL 215, 125, 125, 124,
     120, 112, 96, 0, 63
 370 SYMBOL 216, 255, 255, 4, 6,
     2, 3, 0, 0
 380 SYMBOL 217, 255, 251, 53,
     91, 213, 251, 21, 63
 390 SYMBOL 218, 31, 15, 15, 31,
     31, 31, 31, 31
 400 SYMBOL 219, 255, 255, 255,
     255, 255, 231, 195, 129
 410 SYMBOL 220, 240, 240, 240,
     240, 240, 240, 240, 240
 420 SYMBOL 221, 31, 31, 31, 31,
     31, 31, 31, 31
 430 SYMBOL 222, 129, 129, 195,
     231, 253, 231, 231, 255
 440 SYMBOL 223, 240, 240, 240,
     240, 240, 240, 240, 240
 450 SYMBOL 236, 60, 66, 130,
     192, 192, 224, 127, 63
 460 SYMBOL 237, 252, 126, 71,
     71, 126, 124, 207, 199
 470 SYMBOL 238, 60, 126, 135,
     131, 131, 131, 194, 255
 480 SYMBOL 239, 35, 99, 97,
     101, 109, 109, 109, 255
 490 SYMBOL 240, 60, 62, 39, 35,
     35, 35, 227, 227
 500 SYMBOL 241, 240, 96, 96,
     96, 96, 112, 241, 255
 510 SYMBOL 242, 60, 28, 24, 24,
     24, 28, 28, 254
 520 SYMBOL 243, 62, 30, 12, 24,
     16, 0, 0, 0
 530 SYMBOL 244, 28, 62, 115,
     96, 60, 3, 67, 254
 540 SYMBOL 245, 15, 63, 109,
     115, 115, 109, 63, 15
 550 SYMBOL 246, 240, 252, 134,
     190, 142, 246, 140, 240
 560 SPEED INK 5, 5
 570 INK 0, 0:BORDER 0:INK 1,
     11:INK 2, 24:INK 3, 6:INK
     4, 18:INK 5, 1:INK 6,
     26:INK 7, 13:INK 8, 18,
     1:INK 9, 8:INK 10, 15:INK
     11, 4:INK 12, 7:INK 13, 16
 580 DIM uklad(3)
 590 kredyt=5:bonus=1
 600 ENV 1, 1, 120, 1, 120, -1,
     10
 610 komb(1)=200:komb(2)=200:ko
     mb(3)=200:szansa=0
 620 '**** Ekran ****
 630 MODE 0
 640 DRAWR 638, 0, 3:DRAWR 0,
     398:DRAWR -638, 0:DRAWR
     0, -398
 650 PLOT 0, 380, 4:DRAWR 638,
     0:PLOT 0, 250, 4:DRAWR
     638, 0:PLOT 212, 250:DRAWR
     0, 128:PLOT 428, 250:DRAWR
     0, 128
 660 PLOT 200, 200, 6:DRAWR
     240, 0:DRAWR 0, 30:DRAWR -
     240, 0:DRAWR 0, -30
 670 PLOT 10, 10, 10:DRAWR 400,
     0:DRAWR 0, 80:DRAWR -400,
     0:DRAWR 0, -80:LOCATE
     2, 21:PEN 11:PRINT
     "KOMBINACJA "
 680 PEN 1:FOR n=1 TO
     9:LOCATE 16, n+14:PRINT
     n:NEXT:LOCATE 14, 24:PRINT
     "SZANSA"
 690 LOCATE 2, 15:PEN 13:PRINT
     CHR$(238)+CHR$(241)+CHR$(2
     41)+CHR$(242)+CHR$(243)+C
     HR$(244)+" ";:PEN 7:PRINT
```

Po uruchomieniu programu rozkazem RUN zobaczysz na ekranie komputera przednią ściankę automatu. W górnej części znajdują się trzy pola, na których ukazują się kolejne obrazki: korona, sztabka (ZIP), pistolet, dyskietka, dzwonek i owoc. Na samym dole widać również trzy pola z takimi samymi figurami podpisane KOMBINACJA. O nich za chwilę. Po prawej stronie pojawia się stupek cyfr z napisem SZANSA. Za każdym razem, gdy na środkowym polu wypada korona, migać zaczyna kolejna cyfra. W momencie kiedy miga napis SZANSA i obramowanie ekranu, a nad cyframi pojawia się znaczek „x5”, najbliższe trafienie liczone jest pięciokrotnie! Kiedy zaczyna migać napis RYZYKO i słychać rosnący dźwięk oznacza to, że komputer wybiera szybko cyfry od 0 do 4 znajdujące się pod napisem i po naciśnięciu spacji wybrana cyfra pomnożona przez 1.5 dodana zostanie do wygranej. RYZYKO pojawia się losowo, niezależnie od SZANSY i innych trafień.

Właściwa gra polega na zatrzymaniu zmieniających się u góry obrazków w ten sposób, aby ich układ dał jak największą wygraną. Punktacja jest następująca:

— układ figur identyczny z tym pod napisem KOMBINACJA: Twoja wygrana podwaja się miga następna cyfra przy SZANSIE
— trzy korony: 10 dolarów lub 50 jeśli SZANSA; cyfra przy SZANSIE przeskakuje o dwa
— trzy jednakowe figury: 5 dolarów (25 z SZANSĄ); następna cyfra SZANSY
— korona na środkowym polu: nowa cyfra SZANSY
— dwie korony na skrajnych polach: 2.5 dolara (12.5 z SZANSĄ)
— skrajne pola jednakowe: 2 dolary (10 z SZANSĄ).
— sąsiednie pola jednakowe: pół dolara (2.5 z SZANSĄ).

Zawsze punktowany jest tylko najwartościowszy układ, np. trzy korony nie są już liczone jako trzy jednakowe figury.

Jak zatrzymuje się obrazki (w prawdziwym automacie są to walce). Najpierw automat zabiera jednego dolara z twojego kredytu (na początek otrzymujesz 5 dolarów; aktualna kwota kredytu pokazywana jest w okienku w środku ekranu) i uruchamia środkowy walec. Teraz Ty zatrzymujesz go klawiszem spacji. Ponownym naciśnię-

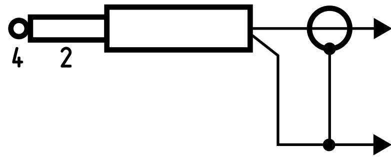
```
     CHR$(236)+CHR$(237)+CHR$(2
     38)+CHR$(239)+CHR$(240)
700 LOCATE 5, 17:PEN 2:PRINT
    "RYZYKO":LOCATE 3, 19:PEN
    6:PRINT "0 1 2 3 4"
710 GOTO 1050
720 **** Symulacja obrotu ****
730 PEN 9:LOCATE 3, 5:PRINT "
    "+CHR$(159)+" ":LOCATE 3,
    6:PRINT " "+" "+" ";:PEN
    6:LOCATE 16, 5:PRINT "
    "+CHR$(159)+" ":LOCATE 16,
    6:PRINT " "+" "+"
740 '
750 a=206:przyp=INT(RND*16)
760 PEN 2
770 LOCATE 10, 5:PRINT CHR$(a)
    + CHR$(a+1) + CHR$(a+2):
    LOCATE 10, 6: PRINTCHR$(a+
    3)+CHR$(a+4)+CHR$(a+5)
780 a$=INKEY$:a$=""
790 IF INKEY(47)=0 OR
    JOY(0)=16 THEN
    uklad(1)=a:GOTO 820
800 IF a<230 THEN a=a+6 ELSE
    a=200
810 GOTO 770
820 nic$=INKEY$:nic$="":IF
    war=0 THEN FOR petla=1
    TO 80:LOCATE 2, 12:PEN
    3:PRINT "START" ELSE 870
830 IF INKEY(18)=0 OR JOY(0)=2
    THEN war=1:LOCATE 2,
    12:PEN 1:PRINT "     ":GOTO
    760
840 IF (INKEY(47)=0 OR
    JOY(0)=16) AND petla>10
    THEN 860
850 NEXT
860 LOCATE 2, 12:PEN 0:PRINT
    "      "
870 FOR pe=1 TO 100:NEXT
880 a=212:war=0:PEN 9
890 LOCATE 3, 5:PRINT CHR$(a
    )+CHR$(a+1)+CHR$(a+2):L
    OCATE 3, 6:PRINT CHR$(a-
    3)+CHR$(a+4)+CHR$(a+5)
900 a$= INKEY$:a$=""
910 IF INKEY(47)=0 OR
    JOY(0)=16 THEN
    uklad(2)=a:GOTO 940
920 IF a<230 THEN a=a+6 ELSE
    a=200
930 GOTO 890
940 FOR pe=1 TO 100:NEXT
950 a=224:PEN 6
960 LOCATE 16, 5:PRINT CHR$(a
    )+CHR$(a+1)+CHR$(a+2):LOCA
    TE 16, 6:PRINTCHR$(a+3)+CH
    R$(a+4)+CHR$(a+5)
970 a$=INKEY$:a$=""
980 IF INKEY(47)=0 OR
    JOY(0)=16 THEN
    uklad(3)=a:GOTO 1250
990 IF a<230 THEN a=a+6 ELSE
    a=200
1000 GOTO 960
1010 '**** Sprawdzenie kredytu
     ****
1020 FOR petla=1 TO 500
1030 IF INKEY$="" THEN NEXT
1040 FOR l=1 TO 50:a$=INKEY$:a
     $="":NEXT:kredyt=kredyt-
     1:PEN 12:LOCATE 8, 12:PRI
     NTUSING"##.##";kredyt;:PR
     INT "$":GOTO 730
1050 IF kredyt<1 THEN LOCATE
     6, 12:PEN 7: PRINT "Gdzie
     forsa!":SOUND 128, 0, 0,
     0:SOUND 1, 20, 100, 5,
     1:SOUND 4, 60, 100, 5,
     1:SOUND 2, 140, 100, 5,
     1:GOTO 1370
1060 IF kredyt>=99.9 THEN
     kredyt=50
1070 PEN 12: LOCATE
     8, 12:PRINT USING
     "##.##";kredyt;:PRINT "$"
1080 FOR t=1 TO 9:LOCATE
     16, t+14:PEN 1:PRINT
     t:NEXT:IF bonus>=10 THEN
     LOCATE 14, 24:PEN 8:
     PRINT "SZANSA": szansa=1:
     bonus=1:ENT ■1, 5, ■1,
     1:SOUND 1, 200, 200,
     5, 0, 1:ELSE LOCATE 16,
     bonus+14:PEN 8:PRINT
     bonus
1090 IF szansa=1 THEN BORDER
     22, 15:ELSE BORDER 0
1100 '**** Kombinacja ****
1110 FOR j=1 TO 3
1120 prz=INT(RND*7):IF prz=1
     THEN komb(j)=200
1130 IF prz=2 THEN komb(j)=206
1140 IF prz=3 THEN komb(j)=212
1150 IF prz=4 THEN komb(j)=218
1160 IF prz=5 THEN komb(j)=224
1170 IF prz=6 THEN komb(j)=230
1180 NEXT
1190 PEN 9:LOCATE 2, 23:PRINT
     CHR$(komb(2))+CHR$(komb(2
     )+1)+CHR$(komb(2)+2):LOCA
     TE 2, 24:PRINT CHR$(komb(
     2)+3)+CHR$(komb(2)+4)+CHR
     $(komb(2)+5)
1200 PEN 2:LOCATE 6, 23:PRINT
     CHR$(komb(1))+CHR$(komb(1
     )+1)+CHR$(komb(1)+2):LOCA
     TE 6, 24:PRINT CHR$(komb(
     1)+3)+CHR$(komb(1)+4)+CHR
     $(komb(1)+5)
1210 PEN 6:LOCATE 10, 23:PRINT
     CHR$(komb(3))+CHR$(komb(3
     )+1)+CHR$(komb(3)+2):LOCA
     TE 10, 24:PRINT CHR$(komb
     (3)+3)+CHR$(komb(3)+4)+CH
     R$(komb(3)+5)
1220 IF szansa=1 THEN LOCATE
     16, 12:PEN 7:PRINT
     CHR$(245)+CHR$(246) ELSE
     LOCATE 16, 12:PEN 1:PRINT
     "  "
1230 FOR l=1 TO
     50:a$=INKEY$:a$="":NEXT
1240 GOTO 1010
1250 '**** Ryzyko ****
1260 IF przyp=5 THEN ENT ■1,
     5, ■1, 1:SOUND 1, 200,
     200, 5, 0, 1:FOR x=1 TO
     200:NEXT:GOSUB 1400
1270 '**** Obliczenia ****
1280 IF szansa=1 THEN kor=50
     :rown=25:skrkor=12.5:sk
     rown=10:obokrown=2.5:LO
     CATE 14, 24:PEN 1:PRINT
     "SZANSA" ELSE kor=10:rown
     =5:skrkor=2.5:skrown=2:ob
     okrown=0.5
1290 IF uklad(1)=komb(1) AND
     uklad(2)=komb(2) AND
     uklad(3)=komb(3) THEN kr
     edyt=(2*kredyt):szansa=
     0:SOUND 1, 25, 100, 5,
     1:bonus=bonus+1:GOTO 1050
1300 IF uklad(1)=200 AND
     uklad(2)=200 AND
     uklad(3)=200 THEN SOUND
     1, 25, 100, 5, 1:szansa=0
     :kredyt=kredyt+kor:bonus=
     bonus+2:GOTO 1050
1310 IF uklad(1)=uklad(2) AND
     uklad(2)=uklad(3) THEN
     SOUND 1, 25, 100, 5, l:sz
     ansa=0:kredyt=kredyt+rown
     :bonus=bonus+1:GOTO 1050
1320 IF uklad(1)=200 THEN
     bonus=bonus+1:SOUND 1,
     25, 100, 5, 1
1330 IF uklad(2)=200 AND
     uklad(3)=200 THEN SOUND
     1, 25, 100, 5, 1:kredyt=k
     redyt+skrkor:szansa=0:GO
     TO 1050
1340 IF uklad(2)=uklad(3) THEN
     SOUND 1, 25, 100, 5, 1:kr
     edyt=kredyt+skrown:szansa
     =0:GOTO 1050
1350 IF uklad(2)=uklad(1) OR
     uklad(1)=uklad(3) THEN
     SOUND 1, 25, 100, 5, 1:sz
     ansa=0:kredyt=kredyt+obok
     rown:GOTO 1050
1360 GOTO 1050
1370 IF INKEY$="" AND JOY(0)=0
     THEN 1370
1380 RUN
1390 '**** Ryzyko ****
1400 war=2:LOCATE 5, 17:PEN
     8:PRINT "RYZYKO"
1410 FOR war=0 TO 8 STEP 2
1420 IF INKEY(47)=0 OR
     JOY(0)=16 THEN 1440
1430 NEXT:GOTO 1410
1440 LOCATE war+2, 19:PEN
     3:PRINT war/2:kredyt=kr
     edyt+(1.5*(war/2)):FOR
     hi=1 TO 700:NEXT:LOCATE
     war+2, 19:PEN 6:PRINT
     war/2:LOCATE 5, 17:PEN
     2:PRINT "RYZYKO"
1450 RETURN
```

ciem klawisza uruchamiasz lewy walec. Naciśnięcie ENTER urucha mia zamiast lewego ponownie środkowy walec. Jeśli nie naciśniesz żadnego klawisza przez 3 sekundy komputer sam uruchomi lewy walec. Lewy walec również zatrzymujesz spacją, co uruchamia automatycznie prawy, który zatrzymujesz analogicznie. Zamiast spacji można użyć dodatkowo przyciskudrążka, a zamiast ENTER pociągać drążek do siebie. Teraz komputer sprawdza, czy coś wygrałeś. Jeśli nie masz już pieniędzy na następną grę automat poinforumuje Cię o tym, lecz nie załamuj się; naciśnij dowolny klawisz i znów dostaniesz 5 dolarów kredytu. Jeśli uda Ci się uzyskać ponad 99 dolarów, to licznik przeskoczy na 50. Możesz wyobrazić sobie, że te pozostałe 50 dolarów właśnie wysypało się z automatu i w nagrodę pójdź do najbliższej kawiarni aby kupić sobie ciastka za 50 złotych. Wesołej zabawy!!!

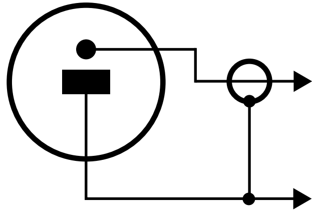
(sw)

JEDNORĘKI BANDYTA

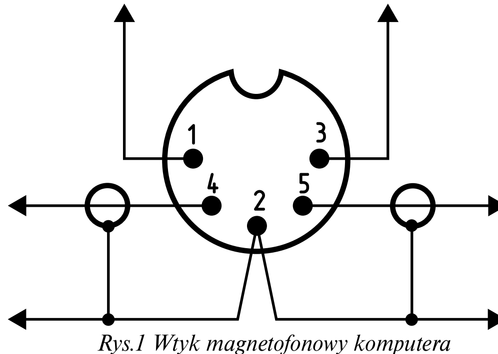
MAGNETOFON I AMSTRAD



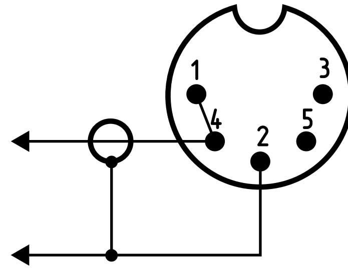
Rys.1 Wtyk słuchawkowy

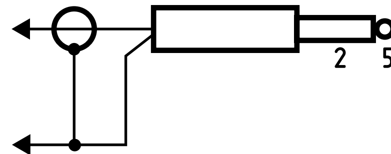


Rys.1 Wtyk głośnikowy



Rys.1 Wtyk magnetofonowy komputera





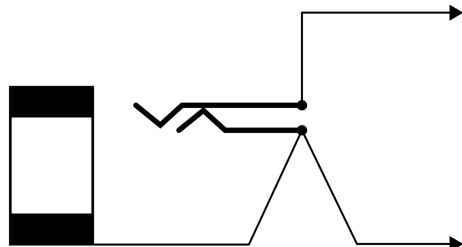
Rys.4, 5 Wtyki mikrofonowe

CPC 6128 i 664 mogą współpracować ze zwykłym magnetofonem. Potrzebny — jest jedynie odpowiedni przewód połączeniowy, którego konstrukcja zależy od typu posiadanego magnetofonu. Różne rodzaje takiego przewodu przedstawia rysunek. Wspólnymi elementami dla wszystkich wersji są: standardowy pięciobolcowy wtyk magnetofonowy współpracujący z gniazdem (w komputerze) oznaczonym „Tape" i cztery przewody o długości od 50 do 80 cm. Dwa z tych przewodów powinny mieć oplot ekranujący. Rozpoczynamy od połączeń wtyku magnetofonowego. Oploty obu przewodów lutujemy do bolca oznaczonego numerem 2, a wewnętrzne przewody odpowiednio do bolców o numerach 4 i 5. Pozostałe, zwykłe dwa przewody lutujemy do bolców o numerach 1 i 3.

Przy wgrywaniu programów z magnetofonu do komputera należy korzystać z gniazda słuchawkowego lub gniazda zewnętrznego głośnika magnetofonu. Uwaga: magnetofony typu „Deck" nie nadają się do naszych celów ze względu na mały poziom sygnału wyjściowego. Jeżeli w magnetofonie znajduje się gniazdo słuchawkowe, to przewód z bolca 4 wtyku magnetofonowego komputera łączymy ze środkowym trzpieniem wtyku słuchawkowego, a oplot tego przewodu z korpusem wtyku (patrz rys. 1). Jeżeli magnetofon posiada gniazdo głośnika zewnętrznego, to przewód z bolca 4 wtyku magnetofonowego komputera łączymy z okrągłym bolcem wtyku głośnikowego, a oplot tego przewodu z bolcem płaskim.

Podobnie postępujemy z wtykiem do nagrywania na magnetofon, z tym, że łączymy przewód z bolca 5 wtyku magnetofonowego komputera z bolcami 1 i 4 wtyku magnetofonowego pięciobolcowego a oplot z bolcem 2 (jeżeli magnetofon ma tego typu gniazdo) lub przewód z bolca 5 wtyku magnetofonowego komputera ze środkowym trzpieniem wtyku magnetofonowego, a oplot przewodu z korpusem wtyku (jeżeli magnetofon ma wejście mikrofoofonowy komputera nowe typu „Jack").

Największym jednak problemem będzie połączenie pozostałych przewodów (bolce 1 i 3 wtyku magnetofonowego komputera), które służą do włączania i wyłączania magnetofonu. Posiadacze magnetofonów z gniazdkiem oznaczonym „Remote" lub „Sterowanie" mają ułatwione zadanie. Wystarczy przylutować te dwa przewody do wtyku typu „Jack" (z reguły o mniejszej średnicy niż wtyk mikrofonowy) pasującego do w/w gniazda. Kolejność tych końcówek jest obojętna. Wszystkim innym użytkownikom pozostaje zainstalowanie dodatkowego gniazda słuchawkowego w magnetofonie i wykorzystanie go do sterowania magnetofonem (patrz rys. 6). Jeden z przewodów doprowadzających zasilanie do magnetofonu należy wylutować i doluto-wać do dłuższego styku wmontowanego gniazda. Następnie dodatkowy przewód podłączyć do korpusu i krótszego styku gniazda, a drugi jego koniec wlutować w miejsce wylutowanego wcześniej przewodu zasilającego. Przeróbka ta dotyczy wyłącznie obwodu zasilania niskonapięciowego. Nie wolno dokonywać takiej zmiany w obwodzie wysokonapięciowym, tj. od strony uzwojenia pierwotnego transformatora zasilacza sieciowego. Należy również zwrócić uwagę na odizolowanie gniazda od metalowych elementów magnetofonu.



Rys. 6 Dodatkowe gniazdo do obwodu zasilania magnetofonu

(wz)

JAK ODBEZPIECZYĆ PROGRAM W BASIC-u

*BASIC komputerów Schneider i Amstrad posiada możliwość zapamiętania programu w formie zabezpieczonej przed listowaniem! Służy do tego rozkaz **SAVE „nazwa"**,P. Tak zapamiętany program można tylko uruchomić rozkazem **RUN „nazwa"** lub **CHAIN „nazwa".** Jeśli zabezpieczyliśmy w ten sposób własny program, to nie będziemy mogli wprowadzać do niego poprawek, ani poprawiać błędów. Aby uniknąć ponownego pisania programu spróbujemy go odbezpieczyć. Pomoże nam w tym krótka procedura w języku maszynowym zamieszczona obok. Ładuje się ona w pusty obszar RAM-u między zmiennymi systemowymi BASICU-u i systemu (B0C7-B0FF) i pozostaje tam nawet po użyciu klawiszy **CTRL/SHIFT/ ESC.** Rozkaz CALL & B0C8 inicjalizuje ją. Od tej chwili wszystkie wczytywane z dyskietki lub taśmy zabezpieczone programy będą traktowane jak zwykłe programy w BASIC-u. Niestety użycie CTRL/ SHIFT/ESC, DISC lub TAPE wyłącza działanie tej procedury. Aby uniknąć ponownego wpisywania jej do pamięci wystarczy przed użyciem powyższych rozkazów wpisać CALL &B0C8, teraz wykonać rozkaz i ponownie wpisać CALL &B0C8.*

Sergiusz Wolicki

```
10 DATA 2177BC11DAB0060
   31A4E777912231310F7C
   9C3ddb0E5D5C5CDC8B0C
   1D1E1CD77BCE5D5C5F5C
   DC8B0E37C3D200167E3F
   1C1D1E1C9
20 s=0:READ a$
30 FOR k=0 TO 51
40 byte=VAL("&" +
   MID$(a$,k*2+1,2))
50 POKE &B0C8+k,byte:
   s=s+byte
60 NEXT k
70 IF s<>7900 THEN
   PRINT "popraw DATA
   !!!":END
80 CALL &B0C8
```

LATAJĄCY KOMPUTER

Zapewne wielu posiadaczy ZX Spectrum zna program Flight Simulation firmy Psion. Możliwości tego programu nie są jednak duże. Widok z kabiny jest nie zbyt urozmaicony, a możliwości sterowania samolotem są bardzo ograniczone. Dla porównania, warto więc zaznajomić się z programem Flight Simulator na komputer IBM PC.

Jak podaje w opisie firma Microsoft, program ten jest symulatorem drugiej generacji. Teren, nad którym odbywają się loty, to całe kontynentalne terytorium USA, część Kanady, Meksyku oraz Karaibów. Oczywiście nawet na największym dysku twardym nie można zgromadzić danych o wszystkich miastach i lotniskach na tym obszarze. Dokładniej zostały przedstawione cztery rejony wielkich miast oraz ponad dwadzieścia lotnisk.

Lot samolotem odbywa się w czasie rzeczywistym, co oznacza, że wydarzenia biegną zgodnie z upływem czasu. Podczas lotu można obserwować deskę rozdzielczą ze wszystkimi przyrządami kontrolnymi (łącznie 28 wskaźników) wymaganymi przez amerykański urząd lotnictwa FAA, trójwymiarowy uproszczony widok z samolotu w jednym z wybranych ośmiu kierunków lub mapę w wybranej skali z zaznaczonym położeniem samolotu. Samolot reaguje na czynności pilota i warunki lotu, tak jak prawdziwy. Przykładowo, przy zbyt ostrym wznoszeniu dochodzi do przeciążenia silnika, zmniejszenia prędkości i samolot traci stateczność.

Podczas lotu możemy zobaczyć takie charakterystyczne budowle jak wieżowiec Empire State Building i most Brooklyn Bridge w Nowym Jorku. Samolot może przelecieć między wieżowcami World Trade Center lub rozbić się o Statuę Wolności.

Przy pewnej wprawie można wykonać takie ewolucje akrobatyczne jak beczka lub pętla. Lot może się odbywać o różnej porze roku i dnia (także w nocy), w różnych warunkach pogodowych (zachmurzeniu, turbulencji, wiatrach na różnych wysokościach). Zbliżając się do lotniska można nawiązać na odpowiedniej częstotliwości kontakt radiowy i uzyskać informacje o pogodzie i warunkach lądowania (w postaci komunikatów na ekranie). Przewidziane są też awarie urządzeń. Przy oblodzenu gaźnika w warunkach zimowych, dla zapobieżenia awarii konieczne jest podgrzanie gaźnika. Jeśli zapomni się wyłączyć światła po nocnym locie, żarówka może ulec przepaleniu. Naprawy dokonuje się po wylądowaniu na lotnisku.

Przeloty między odległymi miastami mogą trwać wiele godzin. Zbiornik paliwa mieści zapas pozwalający na pokonanie 1500 mil. Można także przemieszczać się w przestrzeni zmieniając w przyspieszonym tempie współrzędne położenia w sposób ciągły lub skokowy.

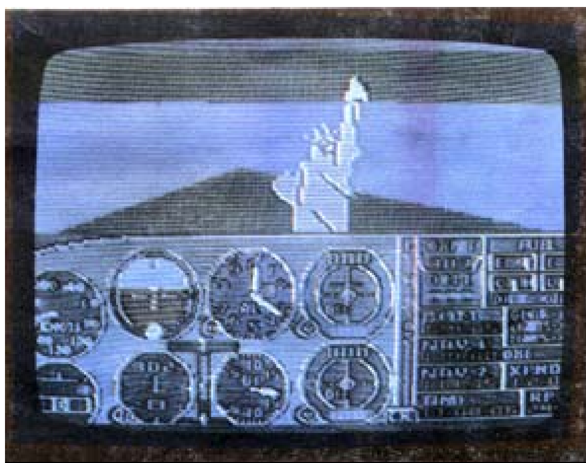
Odrębną opcją programu jest walka powietrzna, w której pilot ma za zadanie zestrzelenie sześciu niemieckich samolotów z okresu pierwszej wojny światowej oraz zniszczenie fabryk zbrojeniowych i zbiorników paliwa.

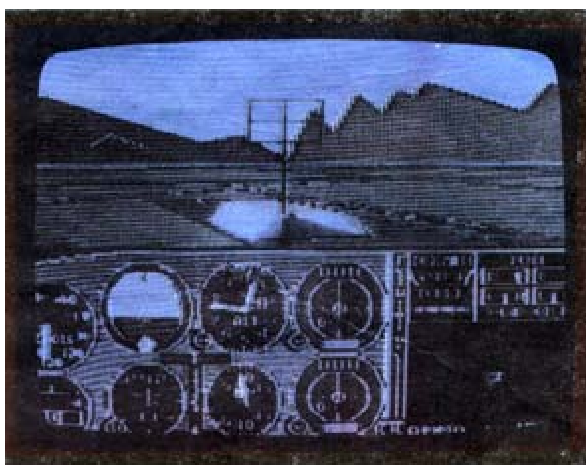
W 1984 roku program został uznany za najlepszą grę na rynku amerykańskim, a obecne wersje są znacznie udoskonalone. Ciekawą sprawą jest to, że pomimo pracy w trybie wysokiej rozdzielczości — w którym zgodnie z instrukcją komputera można uzyskać tylko dwa kolory — widzimy obraz wielobarwny. Autor programu **Bruce Artwick** wyjaśnia tajniki grafiki komputerowej w książce „**Applied Concepts in Microcomputer Grap hics**". Warto dodać, że jest on prezesem firmy komputerowej Sublogic Company specjalizującej się w zastosowaniach graficznych, a wcześniej zajmował się m.in. mikrokomputerowymi systemami sterowania radarów.

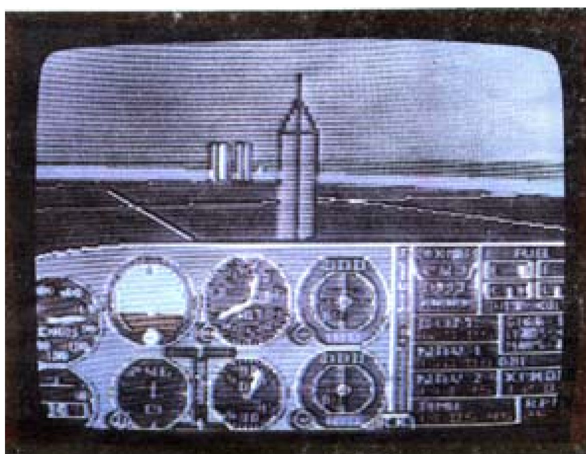
Firma **Microsoft** obiecuje, że **Flight Simulator** pomoże w zrozumieniu zasad pilotażu, ale uprzedza jednocześnie, że takie przeszkolenie nie upoważnia do lotów prawdziwym samolotem.

Michał Szuniewicz









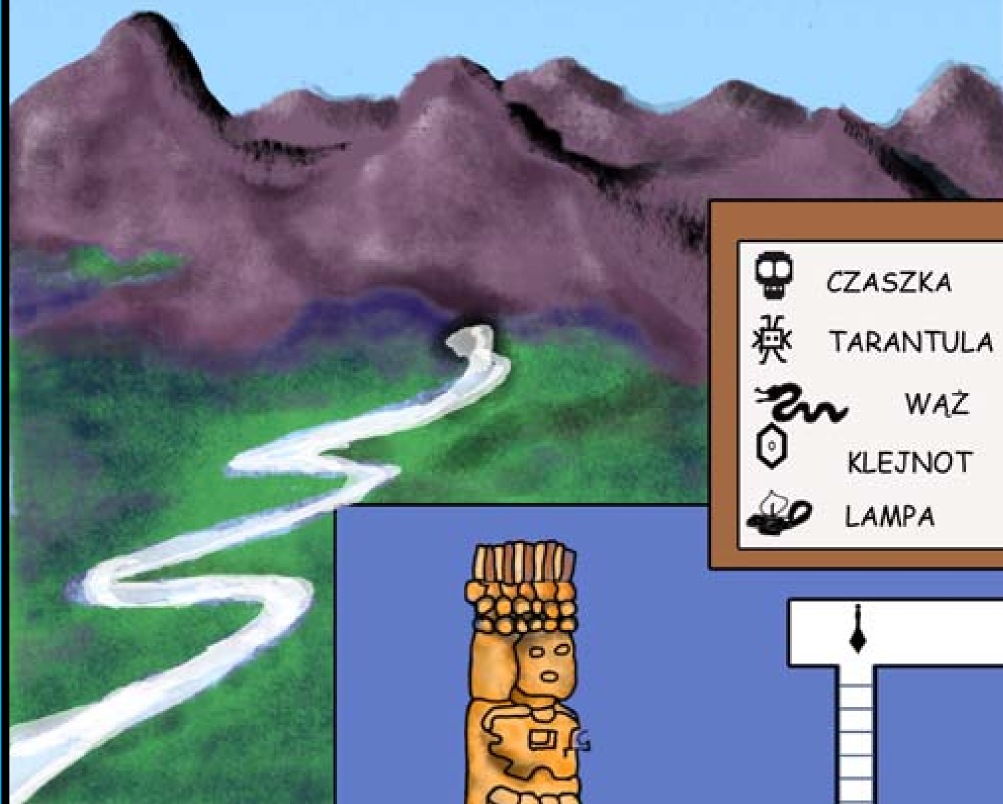
PANAMA JOE

Dawno, dawno temu, w czasach wielkich odkryć geograficznych, kiedy Hiszpanie dotarli do legendarnego królestwa Azteków, krajem tym rządził potężny Montezuma. Na nic jednak zdała się jego potęga wobec zbrojnych wojsk hiszpańskich. Wojsk, które obok krwi żądne były przede wszystkim złota. Złota w państwie Montezumy było pod dostatkiem, traktowane ono było jak najzwyklejszy surowiec do wyrobu ozdób, naczyń i innych niezbędnych przedmiotów. Kiedy mądry władca zobaczył co jest szczególnym przedmiotem pożądania napastników, postanowił ukryć jak największą ilość skarbów, aby nie padły one ofiarą grabieży. I zrobił tak, lecz działał w jak największej tajemnicy. Miejsce, gdzie znajduje się ten skarb przez wieki było nikomu nieznane. Dopiero niedawno udało się odszyfrować Kipu — tajemne pismo węzełkowe, które to ukryli wierni słudzy Montezumy. Według nich skarb ukryty jest w piramidzie zwróconej „twarzą" w kierunku Słońca — odwiecznego boga Azteków. Wielu już śmiałków próbowało odnaleźć nieprzebrane skarby, a wśród nich najcenniejszy posążek Boga Słońca. Nikt jednak nie zdołał wydostać się z labiryntu piramidy.

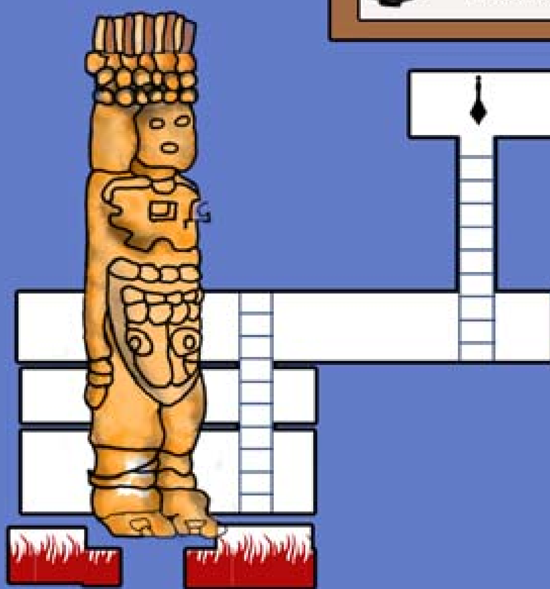
Dzisiaj ty jako Panama Joe podjąłeś się roli poszukiwacza legendarnego skarbca Montezumy. Wędrując przez piramidę nie zapominaj, że nie jest to niedzielna wycieczka, lecz prawdziwa wyprawa pełna przygód i niebezpieczeństw. Na szczęście masz plan piramidy, więc łatwiej Ci się będzie poruszać po ciemnych zakamarkach labiryntu. Czeka tam na ciebie wiele niespodzianek — będziesz musiał stawić czoła żarłocznym pająkom tarantulom, które z upodobaniem atakują wszystkie żywe stworzenia. Jeśli jednak będziesz miał przy sobie znaleziony wcześniej miecz uratujesz jedno ze swoich pięciu istnień, gdyż tarantula ma tylko tyle siły, aby wytrącić ci broń z ręki. Drogę będą ci utrudniały także toczące się czaszki niedoszłych właścicieli fortuny których ta ostatnia wyraźnie opuściła. Możesz je zniszczyć, jeżeli masz miecz. Najlepiej zrobisz, jeśli je po prostu przeskoczysz i pójdziesz dalej w swoją stronę. Może znajdziesz latarnię, która oświetli ci drogę w ciemności? Może znalezione klucze będą pasowały do zamkniętych drzwi? Może fortuna będzie ci przychylna i jako pierwszy i jedyny odnajdziesz wreszcie wymarzony skarb? Twoje szanse są spore, gdyż Panama Joe może odrodzić się aż pięciokrotnie, a jeżeli jego zasługi są odpowiednio wysokie, otrzymuje premię w postaci dodatkowego życia. Może także ustalić sobie stopień trudności jaki będzie w stanie pokonać, łatwy, trudny i bardzo trudny (prawie niemożliwy do pokonania). Jeżeli duch Montezumy będzie ci przychylny i uzna cię za wybrańca Boga Słońca, godnego rozwikłania wielkiej tajemnicy — staniesz się jedynym na świecie posiadaczem ogromnej fortuny, która oprócz bogactwa obdarzy cię czarodziejską mocą zawsze słonecznego uśmiechu.

(ap)

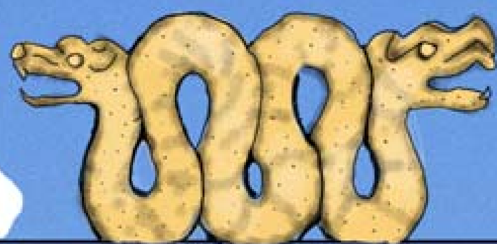




CZASZKA	ŁAŃCUCH	MŁOT
TARANTULA	KLUCZ	MUR NA TRYBIE K I L
WĄŻ	MIECZ	MUR NA TRYBIE J
KLEJNOT	DRZWI	
LAMPA	ZNIKAJĄCA KŁADKA	TRANSPORTER







PANAMA JOE



NIESPODZIANKA

CO JEST GRANE



Tysiące listów i ciągle nowe gry na Atari, Spectrum, Commodore i Amstrada. W sumie czytelnicy głosowali na 831 tytułów.

A oto te najpopularniejsze (czy najlepsze?):

1. IMPOSSIBLE MISSION !
2. ROLAND RAT RACE ⇧
3. FAIRLIGHT !
4. BOMB JACK !
5. SUPERCHESS ⇩
6. ROAD RACE ⇧
7. MATCH POINT ⇩
8. SABOTUR ⇩
9. SKOOL DAZE ⇩
10. THE WAY OF THE EXPLODING FIST

Premię za najlepsze opisy typowanych otrzymuje Piotr Bokina z Gdyni. Nagrodę w postaci zestawu oryginalnych programów na ZX Spectrum ufundowała wypożyczalnia programów **STUDIO XXI** (Hala Koszyki).

Sławek



TOUR DE FRANCE

Trudno dziś znaleźć jakąś dyscyplinę sportową, która nie miałaby swojego odpowiednika w postaci gry komputerowej.

Chcecie się pościgać na rowerach? Proszę bardzo, najbardziej znany wyścig rowerowy, **Tour de France**, czeka na Wasze uczestnictwo. Uprzedzamy – ten szesnastoetapowy wyścig jest naprawdę zabójczy, nie dla nóg jednak lecz dla nadgarstków!

Komu żółtą koszulkę, komu czerwoną? Na starcie stawić się może do 6 kolarzy, którzy jechać mogą przez wszystkie 16 etapów, lub też przez wybrane. Na wzór „Summer Games", każdy uczestnik wystukuje na pięknej, kolorowej palecie, jedną z 16 narodowości, dla której barw narodowych chce zdobyć żółtą koszulkę. Następnie uczestnicy udają się jeden za drugim na linię startu. Niestety, 2 kolarzy nie może jednocześnie wystartować w tym samym czasie: pozostający z tyłu musi się nieźle napracować, aby dogonić leadera.

Krajobraz jest wspaniały, trasa przebiega poprzez trójwymiarowe wzgórza i doliny. Kibice radośnie dopingują. Rower daje się prowadzić lekko. Na długich prostych dodaje się gazu, podczas gdy na zakrętach lepiej jest nieco zwolnić. Nie zbliżajcie się zbytnio do pobocza trasy — upadek kończy się stratą cennego czasu.



Gra ma bardzo ładną grafikę i odznacza się dobrą animacją. Rzeźba terenu, zaznaczona na tablicach informacyjnych oraz różnorodność etapów tworzy z tej sportowej symulacji rowerowej prawdziwą próbę wytrzymałości, w której sam Szurkowski miałby chyba trudności. Jedynym mankamentem Tour de France są długie etapy, które denerwują co bardziej niecierpliwych kolarzy. Może się np. zdarzyć, że na przejechanie 1 etapu potrzeba aż 15 minut! Kto ma jednak nieco cierpliwości, wynagrodzony zostanie wspaniałą grafiką, zaprogramowaną przez bardzo utalentowany zespół węgierskich programistów.

(jz)

PSYTRON

Wprowadzając na rynek grę PSYTRON, firma Beyond uczyniła kolejny krok naprzód w kategorii gier zręcznościowych z rodzaju „zabij go".

Sam scenariusz nie jest zbyt rewelacyjny: musicie obronić kosmiczne osiedle przed atakiem nieprzyjacielskim. Obcy przybysze będą starali się zbombardować budowle, instalacje i wprowadzić niszczycielskie droidy do tuneli.

Jednak to, co odróżnia Psytron od wielu podobnych gier, to niesłychana złożoność i wysokiej klasy grafika. Obraz walki możemy oglądać na 10 ekranach, przedstawiających różne strefy kosmicznej kolonii. Każdy z tych obrazów może być natychmiast przywołany.

Pierwszy poziom może wydawać się łatwy: wszystko co musicie zrobić, to upolować droidy w tunelu. Jeśli osiągniecie pięć razy z rzędu wynik 50% — przechodzicie do następnego poziomu, tj. do zestrzeliwania nieprzyjacielskich statków kosmicznych. W miarę posuwania się do przodu, pojawia się coraz więcej elementów, przeszkód, które musicie pokonać zanim dojdziecie do ostatniego poziomu. Musicie teraz nie tylko stawić czoła najeźdźcom, ale również uważać, aby mała liczba niszczonych przez nich budynków i nie ginęli ludzie. Jeśli np. uszkodzona zostanie zapora — możesz stracić bardzo potrzebne towary.



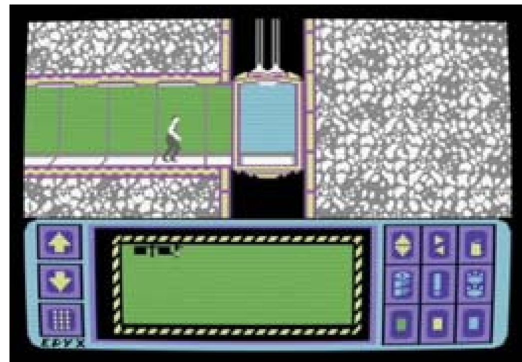
Firma Beyond oferuje Sinclair-a QL każdemu, kto potrafi przeżyć godzinę na najwyższym poziomie. Napiszcie do niej, może się Wam uda. I powiadomcie „Bajtka" — chętnie poznamy zwycięzcę(ów).

Beyond Competiton House
Farndon Road
Market Harborough,
Leicestershire LE 19 9NR
ENGLAND

(jz)

IMPOSSIBLE MISSION by US GOLD

Masz do spełnienia nieprawdopodobną i bezsensowną misję. Musisz, biegając po labiryncie i jeżdżąc windami, przeszukiwać sprzęty umieszczone w tym labiryncie (jest on wybierany losowo). Sprzęty te to krzesła, fotele, lampy, biurka, kanapy. Znaleźć możesz w nich hasło do terminala bezpieczeństwa, kawałek obrazka (puzzle) lub nic. W szukaniu przeszkadzają ci strzelające roboty. Oprócz robotów i sprzętów w pokojach znajdują się ruchome platformy służące do jeżdżenia między poziomami. W pokoju przy wejściu stoi terminal. Jeżeli posiadasz odpowiednie hasło, to możesz uśpić na pewien czas roboty w tym pokoju, lub ustawić platformy w położeniu pierwotnym. Pomiędzy pokojami poruszasz się, jeżdżąc windą. W dwóch z pokoi znajduje się szachownica. Gdy podejdziesz do niej, pokazuje się łapka i na szachownicy rysują się dźwięczące krzyżyki. Twoim zadaniem jest, poruszając łapką, wywołać dźwięki w kolejności od najniższego do najwyższego. Krzyżyki na porządku są trzy, w miarę rozwiązywania zadania ich przybywa. Każda dobrze ustawiona sekwencja krzyżyków daje jedno hasło na uśpienie robotów.



Głównym celem wyprawy jest zebranie i poukładanie z kawałków obrazka. Po zebraniu wszystkich należy poukładać je w jeden obraz na kieszonkowym komputerze. Można je tam obracać, zmieniać im barwę, odbijać „lustrzanie". Komputerek wyposażony jest w urządzenie do porządkowania kawałków i sprawdzania, czy jest ich wystarczająca ilość.

Proste, prawda? Lecz nikt jeszcze nie ukończył tej gry! Wersja na ZX Spectrum wzorowana jest na pierwowzorze, napisanym na Commodore. Pierwowzór ma lepszą grafikę, oprawę dźwiękową, lecz nie ma się co dziwić — na Spectrum gra zajmuje 42kB, a na Commodore ok. 50 kB.

(np)

NA CAŁY EKRAN

Rozpocznijmy od wpisania krótkiego programu:

```
10 DL = PEEK (560) + 256 * PEEK (561)
20 FOR I=0 TO 31
30 PRINT DL+I, PEEK (DL+I)
40 NEXT I
```

Po jego uruchomieniu na ekranie wyświetli się szereg cyfr — jest to tzw. lista obrazu (ang. **d**isplay **l**ist — **DL**). Procesor graficzny Antic pracuje niezależnie od procesora centralnego, ale aby mógł wyświetlać obraz, musi otrzymać informację, jak ten obraz ma wyglądać. Znajduje się ona w tzw. liście obrazu, programie napisanym w specjalnym języku dla procesora **Antic**. Program ten znajduje się w pamięci komputera, a adres jego początku to zawartość dwóch komórek (patrz linia 10). W programie tym możemy wyróżnić pewne Instrukcje wykonywane przez **Antic**:

112 — liczba ta oznacza, że ma być wyświetlonych 8 niewidocznych linii, takich jak w trybie graficznym 8. Trzy te instrukcje dają górny margines obrazu.

66 (64 + 2) — w liczbie tej zakodowane są dwie instrukcje. 64 informuje, że następne dwie liczby w DL to adres pamięci, skąd należy pobierać dane do wyświetlania (początek obszaru pamięci traktowanego jako pamięć obrazu, adres ten możemy też znaleźć sprawdzając komórki 88 i 89); 2 oznacza, że ma być wyświetlona linia w trybie graficznym 0.

Kolejne 23 dwójki to instrukcje wyświetlania 23 linii w trybie 0. Ostatnie trzy liczby stanowią jedną instrukcję i oznaczają, że Antic ma czekać aż nadejdzie pora wyświetlenia kolejnego obrazu; podany jest adres początku display list.

MODYFIKACJE DISPLAY LIST

Modyfikując DL będącą programem dla Antic powodujemy natychmiastowe zmiany w formie wyświetlanego obrazu. Zmieniając którąś z dwójek na 6 możemy sprawić, ze jedna z linii obrazu będzie wyświetlana w trybie graficznym 1, ale ponieważ w tym trybie linia liczy 20 znaków, a w trybie 0, 40 znaków, stąd widoczne przesunięcie części obrazu pod zmienioną linię. Aby można znaczniej zmodyfikować DL przesuńmy ją w inne miejsce pamięci. Nie kasując poprzedniego programu wpiszmy następujące linie:

```
40 POKE 25600+I,PEEK (DL+I)
50 NEXT I
60 POKE 25630,0:POKE 25631,100
70 POKE 560,0:POKE 561,100
```

Teraz przedłużymy DL wstawiając dwie dwójki, czyli spowodujemy, że obraz będzie miało dwie linie poziome więcej, pisząc: (najpierw RESET)

```
75 A=25629
80 POKE A,2: POKE A + 1,2: POKE A + 2,65:
   POKE A + 3,0:POKE A + 4,100
```

Dalsze zwiększanie wysokości obrazu jest niemożliwe, gdyż powoduje widoczne zaburzenia w synchronizacji. Możemy natomiast dodać jeszcze trzy linie u góry obrazu wykorzystując trzy istniejące ale nie wyświetlane linie, zastępując je liniami w trybie graficznym 0. Dopiszmy zatem:

```
110 POKE 88,0:POKE 89,150
120 PRINT „ „
150 A=25600
200 POKE A,66: POKE A + 1,136: POKE A+
   2,149:POKE A + 3,2: POKE A + 4,2: POKE
   A + 5,2
```

Jednak na pewno zauważyłeś, że te pięć dodatkowych linii (trzy na górze i dwie na dole), mimo, że są widoczne na ekranie, to kursor nadal ich nie sięga, pracując w formacie 24 linii. Na to niestety nic nie poradzimy, system operacyjny zarządzający pracą komputera pracuje w tym formacie. Nasuwa się wobec tego pytanie, czy linie te są w ogóle osiągalne, możliwe do zapełnienia. Napisz więc:

```
POKE 38380,10
```

i mniej więcej na środku pierwszej górnej dodatkowej linii pokaże się gwiazdka. Analizując dodane ostatnio linie programu zauważysz, że linia 110 zmienia adres początku pamięci obrazu, linia 200 min. ustala adres pamięci skąd **Antic** ma pobierać dane do wyświetlania. O ile jednak przy normalnym obrazie w trybie gr. 0 początek pamięci obrazu pokrywa się z początkiem pamięci danych do wyświetlania dla **Antic**, to teraz między tymi adresami istnieje różnica 120 bajtów. Stąd właśnie biorą się puste linie u góry (3 x po 40 bajtów). Jeżeli zapiszesz linię 200 następująco:

```
200 POKE A,66: POKE A + 1,0: POKE A
   + 2,150:POKE A + 3,2:POKE A + 4,2:  POKE
   A + 5,2
```

to zauważysz, że obszar roboczy obrazu przesunie się o trzy linie do góry, czyli górne dodatkowe linie będą wykorzystane. Natomiast na dole pojawi się pięć linii, do których nie mamy dostępu kursorem. Wszystko to jest skutkiem organizacji wyświetlania obrazu. Wracając do starej postaci linii 200 — Antic pobiera dane do wyświetlania poczynając od adresu 149 + 256 + 136, trafiając na początku czwartej linii na pamięć obrazu o adresie 150x256, gdy ona się kończy po wyświetleniu 24 linii znów są wyświetla ne puste linie. Dzieje się tak, bo zawartość komórek pamięci, które wykraczają poza pamięć obszaru roboczego obrazu to zera. Zmieniając ich zawartość możesz spowodować, że w liniach tych pojawią się jakieś napisy. Możesz to wykorzystać do dodatkowych opisów programów ale uwaga, dodatkowe dolne linie są przesuwane w górę wraz z całym obrazem, tylko górne linie są niezmienne. Pamiętaj, że zmiany trybów graficznych przywracają normalny stan. Na koniec dodaj jeszcze te trzy linie aby przekonać się jak duży obraz może generować Twój Atari:

```
220 POKE 559,35
400 FOR I=38280 TO 39726
410 POKE I,10
420 NEXT I
```

Na koniec kilka komórek przydatnych przy eksperymentowaniu z obrazem

88,89 — początek pamięci obrazu dla systemu operacyjnego.
82 — lewy margines obrazu w trybie gr. 0 (zwykle 2).
83 — prawy margines (zwykle 39) 93 — kod znaku na pozycji kursora 94,95 — pozycja kursora 85,86 — kolumna, w której znajduje się kursor
84 — linia, w której znajduje się kursor 560,561 — adres display list

Tomasz Kuchn

Klan Atari redaguje Wiesław Migut

KOMPUTER SAM SIĘ PROGRAMUJE

Na to przewrotne pytanie możesz odpowiedzieć wpisując w swój komputer następujący program:

```
10 REM **********************************
20 REM * *
30 REM PROGRAMOWE ROZSZERZENIE *
PROGRAMU *
40 REM  dla ATARI 600/800 XL
50 REM
60 REM
70 REM **********************************
80 PRINT „3”
90 DIM NAZ$ 40
100 PRINT „PODAJ NAZWISKO”
110 INPUT NAZ$
120 PRINT „f”
130 POSITION 2,5: PRINT „300 DATA”; NAZ$
140 PRINT „CONT”
150 POSITION 1,0: PRINT”       „: STOP
170 PRINT”f”
1000 LIST
```

Gdy uruchomisz program komputer zgłosi komunikat:

STOPPED AT LINE 150

Naciśnij i przytrzymaj klawisz RETURN. Gdy kursor przejdzie przez komendę CONT, na ekranie zobaczysz program rozbudowany o linię 300. Pytasz jak to się dzieje to proste na ekranie programowo napisałeś treść linii 300(L. 130), następnie przeniosłeś kursor do góry ekranu (L.150) i zatrzymałeś pracę programu. Naciskając RETURN dopisałeś linię 300 i uruchomiłeś pracę programu od linii 1000. Stosując tę metodę możesz dopisać do programu (lub usunąć) dowolną linię — oczywiście tylko wtedy, gdy jest ona poprawnie zapisana w Atari BASIC.

Sądzę, że przedstawiony tu mechanizm będzie jeszcze bardziej przydatny gdy to nie Ty, ale komputer naciśnie RETURN. By to zrealizować dopisz do programu:

```
145 POKE 842,13:REM włącza RETURN
160 POKE 842,12:REM wyłącza RETURN
```

Jeżeli teraz usuniesz linię 300 i uruchomisz program, przekonasz się, że dopisanie linii odbędzie się automatycznie. Jeżeli chcesz by operacja dopisania do listy była niewidoczna, dopisz kolejne dwie linie:

```
115 SETCOLOR 2,0,0:SETCOLOR 1,0,0:REM czarne litery na czarnym tle
180 SETCOLOR 2,5,5:SETCOLOR 1,10,10:REM zmiana koloru liter i tła
```

Wykorzystanie tego mechanizmu pozostawiam Twojej pomysłowości, bo mądrej głowie... –

Zauważ jednak, że po to by dopisać linię zawierającą instrukcję PRINT nie wystarczy napisać:

```
130 POSITION 2,5:PRINT „300 PRINT”CZY TAK MOŻNA?””
```

By udało się zrealizować to zadanie musisz zapisać:

```
130 POSITION 2,5:PRINT „300 PRINT”; CHR$ (34); „CZY TAK MOŻNA?”; CHR$ (34)
```

Komputer odpowie na to pytanie jeżeli program rozszerzysz o kolejne linie:

```
310 PRINT „MOŻNA PANIE”; NAZS
320 FOR I=1 TO 500: NEXT I: REM PAUZA
```

Przepraszam za antyfeminizm zawarty w linii 310. (Spróbuj zmodyfikować program tak, by rozróżniał płeć operatora.) Powodzenia w eksperymentowaniu.

Zbigniew Górski

PIERWSZY POLSKI KOMPILATOR NA SPECTRUM

Kompilator TOBOS-FP (TOruń BOrkowski Skaba) jest, o ile nam wiadomo, pierwszym całkowicie opracowanym w Polsce kompilatorem zmiennopozycyjnym dla mikrokomputera ZX „Spectrum". Biorąc pod uwagę szybkość działania, zgodność z interpreterem BASICA na ZX „Spectrum" oraz szeroko rozumiana prostotę użycia przewyższa on znane kompilatory produkcji zachodniej takie, jak FP48K-V1.7 czy BLAST.

Przykładowo, **FP48K** nic akceptuje instrukcji **FN i DEF FN**, oraz tablic wielowymiarowych, co go eliminuje z wielu zastosowań. Pozbawiony tych wad **BLAST** jest bardzo długi i kompilacja nawet stosunkowo niedługiego programu jest dosyć czasochłonna, możliwa tylko z wykorzystaniem pamięci masowej. Obydwa kompilatory używają oryginalnych procedur zmiennopozycyjnych, przez co przyśpieszenie wielu programów jest niewielkie lub nawet żadne (patrz test nr 1). Polski kompilator używa własnych procedur zmiennopozycyjnych i zajmuje jedynie 12 K pamięci, dzięki czemu możliwa jest kompilacja nawet 27 K programu w czasie mniejszym niż 1 minuta. Wszystkie obliczenia i wydruki są wykonywane z dokładnością 7 cyfr znaczących.

A oto wyniki pomiaru czasu rysowania funkcji **f(x)=(SQR x)↑3** w przedziale 0-10 za pomocą (programu „Wykresy" opublikowanego w „IKS-ie" nr 1/86. ZX „Spectrum" – 57.5 sek. **FP48K** — 50 sek., oraz **TOBOS -FP** — 4.6 sek. Dla tych, którzy wykorzystują mikrokomputer do opracowywania danych interesującym może być przykład programu dopasowującego wielomian 4 stopnia do 20 punktów doświadczalnych metodą najmniejszych kwadratów. Uzyskane czasy wynoszą:

ZX „Spectrum" — 195 sek. **TOBOS-FP** — 9 sek. Kompilator **FP48K** nie akceptuje tego programu ze względu na użycie tablic dwuwymiarowych. Test nr 2 symuluje ruch sześciu planet wokół Słońca. Dla 100 pętli uzyskano następujące czasy:

ZX „Spectrum" — 165 sek., **FP48K** — 100 sek. oraz **TOBOS-FP** — 9.16 sek. Czasy uzyskiwane za pomocą kompilatora **BLAST** są zbliżone do **FP48K** jednak autorzy nie dysponują obecnie dokładnymi danymi.

TOBOS-FP został w całości napisany w assembierze mikroprocesora Z80 i w postaci źródłowej ma około 7500 linii (instrukcji). Operacje arytmety czne są wykonywane na liczbach w zapisie 4 bajtowym (pojedyncza precyzja) co daje około 7 cyfr znaczących w reprezentacji dziesiętnej. Integralną częścią kompilatora jest tzw. biblioteka procedur arytmetyki zmiennopozycyjnej na którą składają się podprogramy realizujące dodawanie, odejmowanie, mnożenie, dzielenie, funkcje arytmetyczne oraz inne operacje pomocnicze jak np. zapis i odczyt zmiennych. Funkcje arytmetyczne są przybliżane za pomocą tzw. wielomianów Czebyszewa. Przykładowo, funkcję ARCSIN (x) można przybliżyć w przedziale 0-0.5 za pomocą skończonego wielomianu postaci: $W(x) = x + x^3*(-0.5049404/(x^2-3.7042025-1.2516474/(x^2-1.8555182)))$ przy założeniu, że obliczenia są wykonywane na pojedynczej precyzji. Stąd wniosek, że czasy realizacji podstawowych działań arytmetycznych mają zasadniczy wpływ na czasy realizacji funkcji arytmetycznych.

Istnieją dwa zasadnicze rodzaje kodu wynikowego tworzonego przez kompilatory kod naturalny (native code) tzn. kod procesora, na którym będzie realizowany program oraz kod specjalny tworzący zbiór instrukcji tzw. procesora wirtualnego (pozornego) dekodowany w trakcie realizacji programu. Wykonywanie kodu naturalnego jest szybsze lecz zajmuje on znacznie więcej miejsca w pamięci niż kod specjalny. Dlatego autorzy kalkulatora zmiennopozycyjnego zawartego w interpreterze BASIC-a na ZX „Spectrum" stworzyli procesor wirtualny z własnym jednobajtowym zestawem instrukcji. W kompilatorze **TOBOS-FP** zastosowano tzw. bezpośredni kod nizany (**ang. direct threated code**), w którym każda operacja jest reprezentowana przez dwa bajty tworzące razem adres procedury. Skompilowany program stanowi listę adresów poszczególnych procedur przeplataną z danymi a realizowaną przez interpreter wykonujący skoki do kolejnych adresów. Bezpośredni kod nizany jest szybszy od kodu jednobajtowego, gdyż odpada tłumaczenie kodów na adresy, a jednocześnie na tyle oszczędny, że postać źródłowa i wynikowa programu są w przybliżeniu tej samej długości. Poniżej pokazano w jaki sposób jest kompilowana instrukcja LET a=b+c. W podanym przykładzie każda linijka reprezentuje 2 bajty, czyli całość ma 14 bajtów.

FLOAD załaduj liczbę na stos
ADR b adres zmiennej b
FLOAD załaduj liczbę na stos
ADR c adres zmiennej c
FADD dodaj dwie liczby ze stosu, wynik na stosie
FSTOR przechowaj liczbę ze stosu
ADR a adres zmiennej a

Do takiej postaci kodu wynikowego trzeba doprowadzić program źródłowy, aby można go było wykonać. W trakcie kompilacji tekst źródłowy jest wielokrotnie przeszukiwany. Najpierw jest tworzona tablica wszystkich zmiennych, następnie tablica-numerów linii, tablica funkcji i parę innych, po czym są kompilowane definicje funkcji. W dalszej fazie następuje główny przebieg, podczas którego powstaje zasadniczy kod wynikowy. W ostatniej fazie kompilator uzupełnia pewne brakujące fragmenty kodu wynikowego.

Kompilację instrukcji **LET a=b+c** można w skrócie opisać następująco: Kompilator stwierdza, że ma do czynienia z operacją LET, a następnie przechowuje na stosie adres lewej strony wyrażenia. Omijając znak „=" umieszcza na stosie informację o końcu wyrażenia arytmetycznego. Następnie pobiera zmienną b i po wyliczeniu jej adresu kompiluje instrukcję FLOAD b. W dalszym ciągu posyła na stos znak „+", kompiluje instrukcję FLOAD c, po czym stwierdza, że za zmienną c nie ma operacji o wyższym priorytecie niż „+" (np. „*") lub nie ma żadnej operacji, w następstwie czego kompiluje **FADD**. Po otrzymaniu ze stosu informacji o końcu wyrażenia arytmetycznego pobiera przechowany wcześniej adres lewej strony wyrażenia i kompiluje instrukcję **FLOAD**, którą następnie zamienia na **FSTOR**. Powyższy opis jest skrótowy i nie przedstawia różnych istotnych faz kompilacji. Warto zwrócić uwagę, że w ogólnym przypadku mamy do czynienia z operacjami jedno – i dwuargumentowymi, ze zmiennymi różnych typów jak np. zmienne tablicowe oraz z nawiasami zmieniającymi porządek wykonywania operacji.

Wojciech Skaba (lat 22)
Jerzy Borkowski (lat 23)

```
10 LET p2=1.57
30 LET d=0.0
40 LET a=0.0
50 LET s=p2/1000
60 FOR i=1 TO 1000
70 LET a=a+s
80 LET b=SIN a
90 LET c=ASN b
100 LET b=COS c
110 LET c=ACS b
120 LET b=TAN c
130 LET c=ATN b
140 LET b=EXP c
150 LET c=LN b
160 LET b=SQR c
170 LET c=b↑2
180 LET d=d+(a-c)*(a-c)
190 NEXT i
200 LET d=SQR (d/1000)
210 PRINT "dokladnosc =";d
```

Uzyskane czasy: Spectrum – 999 sek., BLAST – 980 sek., FP48K–940 sek., TOBOS-FP – 37.8!!! sek. Dla TOBOS-FP d=1.311E-6.

```
1 LET n=6
5 LET t=.4
6 LET g=-700
7 DIM x(n)
8 DIM y(n)
9 DIM v(n)
10 DIM u(n)
20 FOR i=1 TO n
25 LET x(i)=50*RND+20
26 LET y(i)=58*RND+20
27 LET v(i)=-2
28 LET u(i)=2-RND
30 NEXT i
58 PLOT 120,80
60 FOR j=1 TO 100
61 FOR i=1 TO n
65 PLOT INVERSE
   1,x(i)+120,y(i)+80
66 LET q=x(i)*x(i)*y(i)
67 LET r=9/(q*SQR q)
90 LET ax=R*x(i)
100 LET ay=R*y(i)
110 LET v(i)=v(i)+ax*t
120 LET u(i)=u(i)+ay*t
130 LET x(i)=x(i)+v(i)*t
140 LET y(i)=y(i)+u(i)*t
150 PLOT x(i)+120,y(i)+80
155 NEXT i
160 NEXT j
```

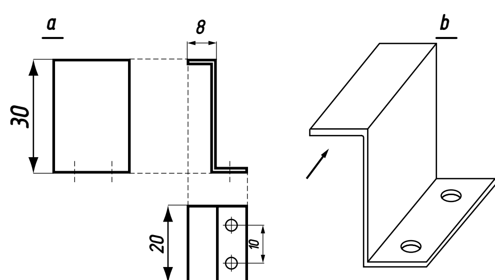
Uzyskane czasy. ZX Spectrum – 165 sek., FP48K - 100 sek. oraz TOBOS-FP – 9.7 6 sek.

Tabela 1

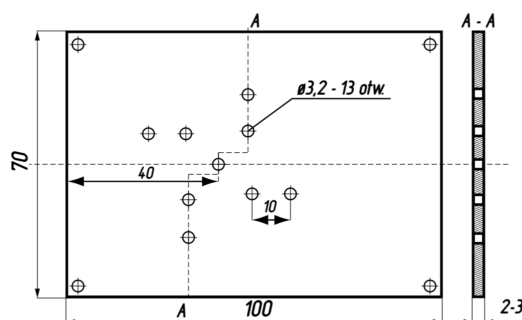
Typowe czasy realizacji niektórych procedur arytmetyki zmiennopozycyjnej podane w milisekundach

Operacja	Czas	Ilość taktów
FMUL	0.40	1400
FDIV	0.64	2240
FADD	0.12	420
FSUB	0.19	665
FSQR	1.35	4725
FEXP	4.3	15050
FLN	3.7	12950
FSIN	3.0	10500
FTAN	5.4	18900
FASN	3.0	10500
FYX	8.5	29750

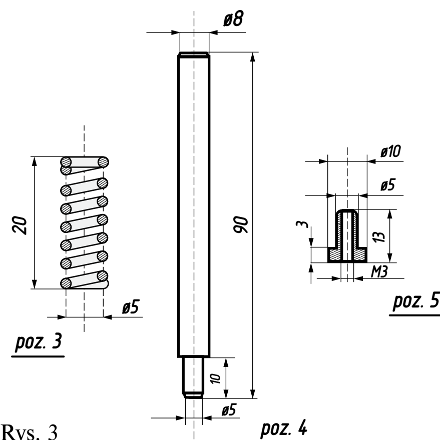
DRĄŻEK STEROWY



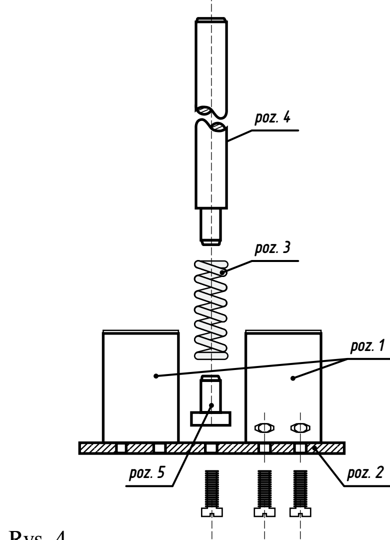
Rys. 1



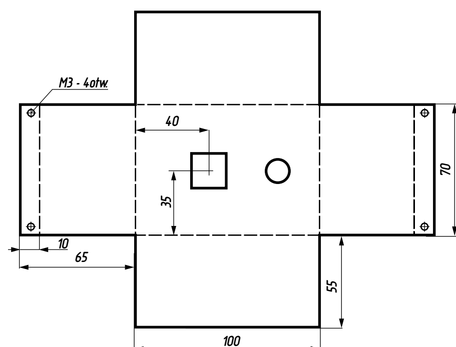
Rys. 2



Rys. 3



Rys. 4



Rys. 5

Jednym z nieodzownych elementów domowego systemu komputerowego jest drążek sterowy. Niestety, dostępne u nas modele mają wiele wad — albo są drogie, albo nietrwałe, albo i jedno i drugie. Dobrym wyjściem jest w tej sytuacji skonstruowanie własnego joysticka.

Wydawałoby się to sprawą prostą, w końcu zasada działania tego urządzenia jest bardzo prosta, jednakże wykonanie zestawu styków w taki sposób, aby był on zarówno wytrzymały jak i wygodny okazuje się dosyć trudne. Autorzy tego artykułu od kilku lat eksperymentują z różnymi modelami drążka z czego najlepszy okazał się model opisany poniżej.

Przed rozpoczęciem pracy należy zgromadzić odpowiednie narzędzia oraz materiały. Potrzebna będzie blacha o grubości ok. 0.8-1.0 mm, płytka z dowolnego materiału izolacyjnego o grubości 2-3 mm, pręt mosiężny o przekroju ok. 8 mm, mocna sprężyna o przekroju wewnętrznym 5 mm i długości ok. 2 cm oraz śruby M3, nakrętki itp. W razie problemów ze zdobyciem materiałów o podanych wymiarach można zastosować inne, podobne, o wymiarach spełniających warunki opisane w artykule.

Pracę zaczynamy od wykonania wszystkich detali. Najpierw zajmiemy się stykami (rys. 1). Na wszystkich rysunkach są one zaznaczone jako poz. 1. Na rys. 1A widzimy sposób wykonania styków, na rys. 1B kierunek siły odkształcającej je. Jak widać, nawet energiczni gracze mogą mieć kłopoty z ich uszkodzeniem. Następnym etapem pracy jest wykonanie płytki. Posłuży ona jako zamocowanie wszystkich elementów. Na rys. 2 nie podano wszystkich wymiarów, ponieważ odległość styków od drążka zależy od wymiarów tych elementów oraz od decyzji konstruktora, czy drążek ma być mniej czy bardziej czuły. Płytka oznaczona jest jako pozycja 2.

Mając wykonaną płytkę i styki można przystąpić do najtrudniejszego etapu — wykonania drążka i jego zamocowania. Elementy te pokazane są na rys. 3. Zasada działania jest następująca: zarówno zamocowanie drążka (poz. 5) jak i sam drążek (poz. 4), wciśnięte są w sprężynę (poz. 3). Dzięki temu drążek zawsze będzie dążył do przyjęcia pozycji pionowej. Dodatkową zaletą tego rozwiązania jest fakt, że gracz wyrywający drążek z pudełka (są tacy) nie spowoduje żadnego szkodzenia — drążek można z powrotem wetknąć na swoje miejsce. Podczas dopasowywania elementów należy zwrócić uwagę na dobre dopasowanie sprężyny do pozostałych elementów. Montaż „na wcisk" powinien zapewnić pewne połączenie. Zmieniając średnicę drążka należy również zmienić średnicę sprężyny oraz jej zamocowania (poz. 5). W razie kłopotów z wykonaniem zamocowania element ten można zastąpić odpowiednio spiłowanym kawałkiem śruby o odpowiedniej średnicy.

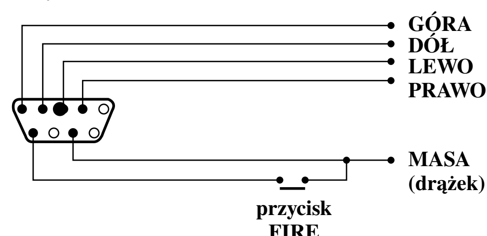
Po wykonaniu wszystkich elementów należy zmontować cały mechanizm i sprawdzić jego działanie. Sposób montażu widoczny jest na rys. 4. Rys. 5 pokazuje przykład wykonania obudowy.

Blachę należy zagiąć o 90 stopni do środka wzdłuż każdej kreskowanej linii. Uzyskane w ten sposób pudełko należy zlutować lub zaspawać. Oczywiście pudełko można wykonać w dowolny sposób, działanie drążka nie jest od tego zależne. Pozostał jeszcze przycisk „FIRE". Potrzebny jest dowolny wyłącznik typu przycisku dzwonkowego. W oryginalnym rozwiązaniu użyto przycisku od elektronicznego klaksonu rowerowego. Przycisk ten „przeżył" prawie rok intensywnego używania, włącznie z turniejami gier komputerowych. Sposób zamocowania przycisku w pudełku zależy zarówno od rodzaju przycisku, jak i od sposobu wykonania pudełka dlatego też pomijamy ten temat.

Jeżeli mamy już zmontowany mechanizm, należy do niego podłączyć kabel. Potrzebny będzie dowolny sześciożyłowy przewód i odpowiednia wtyczka. Nie jest nam znane miejsce, gdzie elementy te można zdobyć, ale zamiast jednego przewodu sześciożyłowego można użyć np. sześciu pojedynczych, a wtyczka — no cóż, myśmy ją wykonali sami. Schemat podłączenia przewodów pokazany jest na rys. 6. Przewody podłączamy po prostu podkładając je pod nakrętki styków i drążka. Poniżej zamieszczamy programy, które pomogą w poprawnym połączeniu przewodów. W C-64 i 128 korzystać należy z portu drugiego.

Michał Silski
Dariusz Rotsztyn

Rys. 6



Widok wtyczki od strony połączeń. Przewody należy podłączyć do odpowiednich styków.

Commodore 64

```
1   DIM D$(10):FOR A=1 TO 10:READ
    D$(A):NEXT
10  X=PEEK(56320):PRINT CHR$(147)
11  IF NOT X AND 16 THEN PRINT
    "FIRE"
12  x=not X and 15: if X>10 THEN
    PRINT "ZWARCIE!":GOTO 10
13  PRINT D$(X):GOTO 10
50  DATA GORA,DOL,ZWARCIE!,LEWO
51  DATA LEWO - GÓRA, LEWO - DOL,
    ZWARCIE!
52  DATA PRAWO, PRAWO - GORA,
    PRAWO - DOL
```

ZX Spectrum

```
1 REM KEMPSTONE TEST
10 DIM A$(11,10): LET S=0
15 LET B$=""
20 FOR T=0 TO 10
30 IF T=3 OR T=7 THEN LET
   A$(T+1)="": NEXT T
40 READ A$(T+1)
50 NEXT T
100 LET A=IN 31: LET S=0
110 IF A>=16 THEN LET A=A-16:
    LET S=l
130 PRINT AT 10,12;A$(A+1)
140 IF S=1 THEN LET
    B$="STRZAL"
145 PRINT AT 11,12;B$
150 IF S=0 THEN LET B$="       "
160 GO TO 100
1000 DATA "SRODEK","PRAWO"
1010 DATA "LEWO","DOL"
1020 DATA "PRAWO-DOL"
1030 DATA "LEWO-DOL"
1040 DATA "GORA","PRAWO-GORA"
1050 DATA "LEWO-GORA"
```

WARTO PRZECZYTAĆ



Micro STRAD

Wraz z napływem komputerów Amstrad do naszego kraju, pojawiają się też — choć w wolniejszym tempie — różne czasopisma poświęcone temu komputerowi. Jednym z nich jest francuski „Micro STRAD".

Wydawany przez koncern prasowy „Groupe tests" dwumiesięcznik „**Micro STRAD**" reklamuje się jako pismo całkowicie niezależne od firmy Amstrad. Niezależnie i — powiedzmy to od razu — jedno z najdroższych.

Pismo — oprócz okładki — jest czarno-białe, drukowane na kredowym papierze, znacznie podnosi czytelność drukowanych programów. Nie oznacza to bynajmniej, iż „**Micro STRAD**" zamieszcza na swoich łamach głównie wydruki komputerowe, oprócz krótkich, syntetycznych artykułów dotyczących różnych urządzeń peryferyjnych i oprogramowania, znajdujemy krótkie mikro-plotki o nowościach, głównie z drugiej strony kanału La Manche i RFN.

Przeglądając ostatni numer omawianego czasopisma natrafimy na szereg ciekawych artykułów, a m.in.: „PCW 8256 od podszewki", „Szara myszka AMX", „O muzyce na Amstrada", „Textomat i twój tekst", „Jak zaadaptować programy CPC 664 na CPC 464" i wiele innych. Redakcja wpadła na ciekawy pomysł: na osobnej stronie opublikowane zostały adresy wszystkich cytowanych w całym numerze firm produkujących różnorodne urządzenia peryferyjne lub akcesoria dla Amstrada, i to nie tylko francuskich, ale również angielskich i niemieckich.

Na uwagę zasługuje sposób oceny 20 programów komputerowych przy pomocy wykresów, wpisanych między osie współrzędnych, przy czym ocenie poddane zostały następujące kryteria: łatwość w posługiwaniu się programem, grafika, dźwięk, szybkość programu i... ogólna ocena redakcji. Najlepszy wynik osiągnęły programy: 3d Voice Chess, Death Pit, Infernal Runner.

Numer zamyka ocena kilku najnowszych podręczników programowania oraz kilka „sposobów i sposobików", m.in. jak uniknąć błędów w programowaniu, jak dostosować każdy magnetofon do Amstrada itp.

„Micro STRAD", 5, Place du Colonel Fabien, 75491 PARIS Cedex 10



SINCLAIR USER

Chyba najbardziej znanym w Polsce pismem komputerowym poświęconym komputerowi ZX-Spectrum jest wydawany w Wlk. Brytanii miesięcznik SINCLAIR USER.

SINCLAIR USER należy do typu czasopism komputerowych poświęconych produktom tylko jednej firmy, w tym wypadku komputerom (dziś już bankruta) papy Clive Sinclaira. Na prawie 150 stronach znajdujemy już tradycyjnie ten sam rodzaj artykułów i ogłoszeń, na jakie możemy się natknąć, przeglądając inne, podobne czasopisma. Pod tym względem **SINCLAIR USER** nie różni się zbytnio np. od „**AMSTRAD USER**". Z jednym wyjątkiem: pismo publikuje adresy brytyjskich i światowych klubów użytkowników ZX-Spectrum oraz aktualną listę nowych programów dla ZX-81, Spectrum 16k i 48k, biorąc pod uwagę różne kategorie, np. gry przygodowe, zręcznościowe, programy edukacyjne, programy użytkowe itp. Jest to praktyczny, aktualizowany miesięcznie przewodnik po nowościach, jakie w dalszym ciągu ukazują się w niezliczonych ilościach na ten popularny w Europie komputer.

Czytelnik znajdzie w opisywanym piśmie różne artykuły teoretyczne, dotyczące budowy i działania ZX-81, ZX-Spectrum i jego odmian, wywiady, przegląd nowych urządzeń peryferyjnych dla wspomnianych modeli, a także recenzje najnowszych gier napisanych na Spectrum. Jak zwykle redakcja nie zapomniała o zapalonych programistach, którzy lubią rozgrzewać sobie palce na klawiaturze: dla nich właśnie dedykowana jest kolejna porcja wydruków programów. Przegląd najnowszych książek i podręczników, kącik programów czytelników, wskazówki dla początkujących programistów — oto pozostałe, stałe rubryki SINCLAIR USER.

SINCLAIR USER jest popularnym, nteresującym, bogato ilustrowanym pismem, przydatnym dla każdego posiadacza komputera ZX-81 lub ZX-Spectrum. Do jego wad zaliczyć można zbyt dużą liczbę reklam, zajmujących sporą część objętości. Ale jest to powszechnie spotykana wada większości czasopism komputerowych, których żywot uzależniony jest nie tylko od ilości sprzedanych egzemplarzy, ale i od wpływów uzyskanych za same ogłoszenia i reklamy.

„SINCLAIR USER", ECC Publications. 196-200 Balls Pond Road, London NI 4 AO.



CHIP

CHIP — to tytuł najpoważniejszego zachodnio-niemieckiego miesięcznika komputerowego, wydawanego przez jeden z największych koncernów prasowych, **Vereinigte Motor Verlage GmbH**.

CHIP jest zbliżony do amerykańskiego pisma **BYTE**. Obszerne, ponad 300 stronicowe czasopismo, przynosi każdego miesiąca obfitą dawkę informacji, od artykułów problemowych poprzez testy komputerów kompatybilnych z IBM AT (już nie PC!) oraz urządzeń peryferyjnych (np. drukarek), do sporej części ogłoszeń drobnych i różnego rodzaju reklam.

Przeglądając, nawet pobieżnie, **CHIP**-a, uderza nas przede wszystkim wysoki profesjonalizm i duży poziom edytorski. Pismo ma staranną szatę graficzną, duża liczba drobnych ogłoszeń przeplatana jest reklamami handlowymi.

Od pewnego czasu redakcja **CHIP** reklamuje również specjalne wydania swojego czasopisma, poświęcone poszczególnym komputerom. Dzięki specjalnej karcie zamówieniowej, dołączonej do każdego numeru, możemy zamówić tytuły: „AMIGA dla początkujących", „Turbo Pascal" część I i II, „MSX-2 dla każdego" itp.

CHIP nie jest pismem łatwym w odbiorze, nawet nie tyle ze względu na specjalistyczny język, co z uwagi na ścisłe ukierunkowanie profesjonalne; choć publikuje informacje potrzebne użytkownikom popularnych komputerów — C 64, TI-99, ZX-Spectrum i innych — to jednak głównie jest ono nastawione na czytelnika wyrobionego, mającego na codzień kontakt z komputerami klasy IBM.

A więc „**BYTE**" czy „**CHIP**"? Wszystko zależy od tego, czego kto szuka. No i jaki obcy język zna lepiej.

„CHIP" Schillerstrasse 23a, 8000 München 2

LASER 128

Nareszcie mamy szansę skorzystania z największej biblioteki oprogramowania, jaka kiedykolwiek powstała dla mikrokomputerów. Od niedawna można w Polsce kupić komputer **LASER 128,** znakomicie podtrzymujący tradycje **Apple II** — pierwszego mikrokomputera domowego i jedynego, którego powodzenie utrzymuje się od lat na jednakowo wysokim poziomie. Konstrukcja LASERa i jego możliwości użytkowe kwalifikują go do klasy mikrokomputerów profesjonalnych. Oto szczegóły.



Najważniejszym parametrem jest zgodność z **Apple IIe**. Zbudowany na mikroprocesorze **6502** w wersji **CMOS**, pracuje z zegarem o częstotliwości 1 MHz. tak jak wszystkie **Apple II**. Dysponuje 128 kB pamięci operacyjnej RAM, z możliwością rozszerzenia do 256 kB. 32 kB pamięci stałej ROM zawiera interpreter BASIC-a standartu Microsoft, maszynowe procedury obsługi terminala, klawiatury, dysków i wszystkich wbudowanych interfejsów. Nasz bohater jest bowiem od razu wyposażony w interfejsy do typowego wyposażenia:

— wejście (wyjścia równoległe standartu **CENTRONICS**, przeznaczone dla drukarki lub plotera,
— dwa wejścia (wyjście szeregowe **RS232C**, jedno przeznaczone do przyłączenia modemu służącego do transmisji danych na większe odległości, lub połączenia bezpośrednio z drugim komputerem, a nawet do przyłączenia dodatkowej klawiatury,



a drugie — do drukarki lub plotera z wejściem szeregowym. W obu przypadkach można korzystać z typowych szybkości — od 110 do 19200 bodów;
— port dla zewnętrznej stacji dysków, drugiej z kolei, ponieważ pierwsza jest na stałe wbudowana w komputer — ale o tym za chwilę;
— wyjście o rozdzielonych sygnałach wizji, koloru i synchronizacji — dla monitora **RGB**, monitora o zmiennej jasności (takiego jak w IBM), lub dla ekranu ciekłokrystalicznego (LCD);
— wejście zespolonego sygnału wizyjnego do dołączenia monitora monochromatycznego lub kolorowego w standarcie PAL (lub innym — na zamówienie), niestandartowej komunikacji z komputerem;
— port dla myszy, drążków sterowych i innych manipulatorów używanych do gier lub transmisji
— wyjście sygnału z regulacją poziomu, typowo sterującego głośnik, z możliwością generowania od 0 do 250000 impulsów na sekundę.

Na zewnątrz komputera jest również wyprowadzona jego magistrala systemowa, dokładnie taka sama jak w Apple II, co umożliwia wykorzystanie modułów Apple do rozbudowy **LASERa**. Są też do nabycia moduły przeznaczone specjalnie do niego. Wśród nich do najbardziej atrakcyjnych należą: pakiety z procesorami **Z80, 8086, 6809** i **68000**, kontroler z dyskiem **WINCHESTER**, syntezatory dźwięku i mowy, interfejsy do sterowania w układach automatyki, przetwornik do wprowadzania obrazu telewizyjnego, zegar czasu rzeczywistego i wiele, wiele innych.

Olbrzymią zaletą **LASERa** 128 jest wbudowana stacja dysków, o popularnym standarcie 5¼ cala, pracujący w podwójnej gęstości zapisu. Takie rozwiązanie pozwala na zawsze zapomnieć o istnieniu magnetofonu kasetowego.

Klawiatura **LASERa** ma wydzielony blok klawiszy numerycznych i 10 klawiszy funkcyjnych. Układ, wyprofilowanie i wielkość klawiszy — jak w rozwiązaniach profesjonalnych. Standard ekranu charakterystyczny dla dobrego komputera — 40 lub 80 znaków w 24 liniach, lub grafika od 40 x 48 do 560 x 384 punktów w sześciu kolorach.

Komputer jest niewielki (37 x 31 x 8 cm), lekki i wyposażony w rączkę do przenoszenia. Przy zastosowaniu ekranu **LCD** i zasilania z baterii daje to możliwość pracy w terenie, lub podczas podróży.

Na temat oprogramowania do **Apple II**, w całości dostępnego dla użytkownika **LASERa 128**, pisane są całe książki. Poważne firmy software owe rozprowadzają grube katalogi i katalogi tych katalogów. Całą tę masę można podzielić na następujące grupy:

— systemy operacyjne,
— języki programowania,
— programy narzędziowe
— procesory tekstu,
— oprogramowanie biurowe i wspomagające zarządzanie
— programy edukacyjne,
— gry.

LASER 128 może pracować pod każdym systemem operacyjnym przeznaczonym dla **Apple II** — za równo pod dowolną z kilkunastu wersji **Apple-DOS**, jak i pod **CP/M 2.2**, adaptowanym z **PDP 11** systemem **UCSD** i (po uzupełnieniu o moduł z procesorem **8088**) pod **CP/M 86** lub **MS-DOS**. Dzięki tym dwóm ostatnim **LASER 128** umożliwia wykorzystanie niektórych programów z **IBM PC**! Jeżeli chodzi o języki programowania, to jest w czym wybierać: **Basic, Pascal, Fortran, Fortran 77, Ada, Modu-la 2, PL/1, Algol, Lisp, Cobol, Logo, Forth, C, Prolog, Pilot**, asemblery i wiele innych, mniej znanych. Należy podkreślić, że w przeciwieństwie do wielu implementacji mikrokomputerowych, są to pełnowartościowe translatory języków programowania wraz z odpowiednimi środowiskami programowymi, umożliwiające tworzenie zaawansowanego oprogramowania aplikacyjnego.

Zaawansowane procesory tekstu (**Wordstar**,) typowe programy do obliczeń planszowych (**Visicalc**), bazy danych (**dBASE**) i pakiety zintegrowane (**Appleworks**) gwarantują przydatność **LASERa 128** w każdym biurze i na biurku każdego dyrektora. Znane i cenione na całym świecie oprogramowania edukacyjne, zawierające nawet współpracujące z myszą pakiety pomagające przy rehabilitacji osób niepełnosprawnych, dopełnia obrazu komputera przydatnego praktycznie wszędzie.

Wśród popularnych u nas mikrokomputerów należy pod względem możliwości wykorzystania i parametrów użytkowych umiejscowić **LASERa 128** pomiędzy **Amstradem** a **IBM PC**, na pewno bliżej **IBM.** Dobrze, że pojawił się na naszym rynku, bo jest on znacznie tańszy od IBM, a częstokroć może go z powodzeniem zastąpić, szczególnie w pracy biurowej. Niepodważalnymi atutami są zacznie mniejsze wymiary, brak wewnątrz obudowy hałasującego wentylatora, powodującego nerwice u użytkowników **IBM** i o wiele łatwiejsza obsługa. **LASER 128** nie wymaga od użytkownika znajomości zawiłości komputera i systemu operacyjnego w tym stopniu, co **IBM PC** i umożliwia bardzo łatwe i efektywne tworzenie oprogramowania na własny użytek.

Maciej Kasperski

DOBRY PROGRAM

musi spełniać bardzo wiele warunków, aby na dobra ocenę zasłużyć. Musi być szybki, łatwo modyfikowalny, zajmować możliwie najmniej pamięci.

Wymieniłem tylko kilka warunków, najczęściej stawianych w podręcznikach programowania, czyli książkach pisanych przez programistów dla programistów. Wszystkie te i podobne do nich, kryteria są słuszne i muszą być spełnione, jednak należy do nich dołączyć wymagania użytkownika, dla którego dobre programy to przede wszystkim te,z którymi współpracuje się sprawnie i przyjemnie.

Jeśli człowiek chce być lubiany, musi być dla innych ludzi miły i uprzejmy. Jeśli program ma odnieść sukces, czyli uzyskać szerokie grono użytkowników, także musi się zachowywać zgodnie z pewnymi zasadami.

Jak ma wyglądać zbiór zasad dobrego wychowania dla programów? Jednoznacznej odpowiedzi na to pytanie nie ma, bo to zawsze trochę rzecz gustu — jedni ludzie wolą towarzystwo rozmowne, inni małomówne, itd. Wydaje mi się jednak, że pewne podstawowe zasady można spróbować sformułować i dlatego postanowiłem przygotować malutki podręcznik, z którego programy będą mogły się uczyć dobrych manier. Oto on:

DOBRE MANIERY DLA PROGRAMÓW

Po pierwsze, przedstaw się człowiekowi, z którym będziesz pracował. Oprócz swojej nazwy, która być może nic mu nie powie, napisz kilka słów o sobie — w czym możesz mu pomóc. Możesz też napisać kiedy się urodziłeś i kto jest Twoim ojcem. Ale nie powinieneś być zbyt gadatliwy — może masz przed sobą kogoś, kto już Cię doskonale zna i szkoda mu czasu na czytanie tego samego po raz setny. Za to na specjalne życzenie powinieneś potrafić opowiedzieć o sobie coś bliższego.

Skoro już prezentacja została dokonana należy omówić szczegóły zadania, które masz wykonać. Może jest to rzecz całkiem prosta, np. rozwiązanie równania. Wtedy szybko poproś o podanie współczynników, oblicz i podaj wyniki. Ale nie uciekaj od razu do systemu operacyjnego. Zapytaj czy jest jeszcze coś do zrobienia i dopiero gdy wszystkie potrzebne równania będą rozwiązane możesz grzecznie powiedzieć „Do widzenia".

Zawsze bardzo uważnie słuchaj co (za pośrednictwem klawiatury) mówi Twój partner. Jeśli coś wyda Ci się podejrzane, poproś o powtórzenie. Np. jeśli spodziewasz się podania liczby a otrzymałeś znaki: 1o (jedynka i litera o) nie możesz wypisać „SYNTAX ERROR" i uciec bez pożegnania, zostawiając osłupiałego rozmówcę z interpreterem BASIC-a, powtarzającym swoje „READY" z uporem dziecka grającego w pomidora. Należy raczej uprzejmie zasugerować:

Błędnie podana liczba, proszę powtórzyć
i kontynuować pracę.

Pamiętaj też, że jako ekspert od wykonywania swojego zadania powinieneś wykrywać także bardziej skomplikowane błędy w danych. Np. jeśli wczytujesz długość i szerokość prostokąta, to wolno Ci przyjąć tylko liczby większe od zera. W przeciwnym przypadku zwróć uwagę użytkownikowi, że chyba się pomylił. I oczywiście nie „Do widzenia", tylko poproś o poprawne wartości.

Czasami zdarza się, że użytkownik podłącza do Twojego komputera drukarkę i prosi o zanotowanie wyników na papierze. Nie zapomnij przed podaniem wyników wypisać najpierw danych, które podał. Jeśli tego nie zrobisz, a będziesz rozwiązywał kilka zadań, to na papierze znajdzie się kilka wyników, które będą zupełnie bezwartościowe, bo człowiek (takie są niestety zwyczaje ludzi) zapomni, które wyniki są dla których danych! I trzeba będzie wszystko liczyć od nowa. Mało tego, chociaż to jego wina — mógł nie zapominać! — to będzie z Ciebie bardzo niezadowolny. Więc napisz mu dla świętego spokoju te dane i jeszcze kilka słów objaśnienia co właściwie oznaczają wydrukowane liczby. Może to wyglądać np. tak:

Pole prostokąta o bokach a = 5, b = 6 wynosi 30 (5 i 6 to dane, które podał, 30 wynik działania programu).

Jeśli Twój komputer ma kalendarz i zegarek, to czasami warto na początku lub na końcu wydruku napisać aktualną datę i godzinę — ludziom potrafią się przydać najdziwniejsze rzeczy. (O tym, że na początku wydruku trzeba się przedstawić nie muszę Ci chyba przypominać).

Co prawda z tymi wydrukami to trzeba bardzo uważać. Jeśli jesteś np. specjalnym programem do pisania listów, czyli Twoje wydruki to będą listy wysyłane później do innych ludzi. To chyba użytkownik będzie wolał żeby na papierze nie zostawały Twoje wizytówki.

Bardzo często zdarza się, że jesteś programem wszechstronnym i potrafisz robić wiele różnych rzeczy. W takim wypadku bądź również uczynny — zaoferuj od razu wszystkie swoje możliwości. Możesz np. wypisać na ekranie:

POTRAFIĘ OBLICZYĆ

A — objętość kuli
B — objętość sześcianu
C — objętość stożka
Co mam zrobić dla Ciebie?

To ostatnie pytanie jest w zasadzie uprzejme, ale i kłopotliwe, bo teraz człowiek nie wie jak ma powiedzieć, że chciałby akurat stożek. Więc jak zapytać?

Pomoże nam tutaj następna ogólna zasada: nie daj się długo prosić! Pozwól, aby użytkownik wydawał Ci polecenia jak najprościej, np. jednym naciśnięciem klawisza. W naszym ostatnim przykładzie, po wyświetleniu oznaczonych literami możliwości, zamiast pytania „Co mam zrobić dla Ciebie?", możesz zaproponować:

Wciśnij wybraną literę.

Inna możliwość usprawnienia komunikacji, to wyświetlenie dostępnych czynności, a użytkownik niech wybiera przez naprowadzenie kursora na jedną z nich. Może to wyglądać np. tak: piszesz „OBLICZANIE OBJĘTOŚCI" i wyświetlasz symboliczne rysunki brył. dla których potrafisz

policzyć objętość. Kursor ustawiasz pod pierwszym z nich. Użytkownik naciska na klawiaturze strzałkę — przesuwasz kursor pod następny obrazek. Gdy już kursor znajdzie się pod właściwą bryłą, użytkownik, naciskając np. RETURN, poinformuje Cię, że to jest jego wybór.

Jesteś już uczynny i gotów do pracy na każde skinienie? Tak. To świetnie. A pamiętasz niedźwiedzią przysługę, czyli historyjkę o misiu, który z nadmiaru uczynności rozbił przyjacielowi głowę? No właśnie, z uczynnością trzeba bardzo uważać. I to właśnie Ty masz uważać. Twój partner to przecież Człowiek, istota omylna, która czasami nawet nie potrafi dobrze trafić palcem w klawisz. Jeśli widzisz, że prosi Cię o coś co może spowodować straty (np. cały wieczór pisał list, a teraz nacisnął klawisz oznaczający „skasuj wszystko, zaczynamy od początku"), to warto upewnić się, że to nie pomyłka, tylko świadomy krok.

Nie myśl tylko, że namawiam Cię do nieposłuszeństwa. Jeśli partner prosi Cię o coś i stanowczo tę prośbę podtrzymuje, to musisz ją wykonać bez wahania. Chyba, że jest niewykonalna. Np. dzielenie przez zero, lub wyciąganie pierwiastka kwadratowego z ujemnej liczby. Ale w takiej sytuacji trzeba go poinformować: **To się nie da zrobić!!!.**

Następne cechy bardzo potrzebne w kontaktach z ludźmi to cierpliwość i wyrozumiałość. Byliśmy już niedaleko od nich mówiąc o korygowaniu błędnych danych. Będą Ci nieraz potrzebne także w innych okolicznościach. Skoro ludzie z niezwykłą łatwością o wszystkim zapominają, to nietrudno przewidzieć, że nieraz w trakcie pracy Twój partner zapomni jak ma wydawać Ci polecenia, albo w ogóle co jeszcze potrafisz dla niego zrobić. Najlepiej by było gdyby mógł wtedy napisać „**POMOCY**", lub po prostu wcisnąć klawisz „?" i w odpowiedzi otrzymać krótką informację o tym co może od Ciebie uzyskać i w jaki sposób.

Właściwie doszliśmy już do końca naszego samouczka. Jeśli nie przypadł Ci do gustu, bo na przykład nie lubisz stosować się do sztywnych reguł i chcesz kierować się tylko własnym rozumem (to się często zdarza dorastającym programom), to może chociaż zaakceptujesz przynajmniej jedną ogólną zasadę: „PROGRAMY SĄ PO TO, ŻEBY LUDZIOM BYŁO WYGODNIEJ, a nie na odwrót".

Powyższy zbiorek, trafił do „Bajtka", bo choć adresowany do programów, może dostarczyć pożytecznych informacji także naszym Czytelnikom. Mam na myśli zarówno tych, którzy nastawiają się raczej na korzystanie z gotowego oprogramowania, jak i tych, którzy sami chcą oprogramowanie tworzyć.

Użytkownicy powinni sobie zdawać sprawę czego należy wymagać od dobrych programów. Choćby po to żeby nie dać sobie „wcisnąć" byle czego. Nawet w sytuacji gdy rynek oprogramowania jest bardzo ubogi i trudno zdobyć jakikolwiek program, warto mieć wysokie wymagania, bo właśnie wymagania użytkowników są jednym z czynników określających co znajdzie się na rynku za rok czy dwa lata.

Natomiast programiści muszą pamiętać, że na nich, jako na „rodzicach" spoczywa obowiązek wpojenia programom zasad dobrego tonu. To, czy dokład-

nie takich jak wyżej wymienione, może być przedmiotem dyskusji, wiele zależy od konkretnego przypadku. Nasz zbiór reguł można potraktować jako podstawę do dyskusji lub do stworzenia podobnego, na użytek własnych programów.

Na zakończenie chciałbym przedyskutować wątpliwości jakie może budzić postulat tworzenia programów: „życzliwych dla użytkownika". Przecież napisanie takiego programu wymaga dużego nakładu pracy. Czy nie jest to praca zmarnowana? Przecież jeśli napiszę sobie program, to doskonale wiem jak z niego korzystać

Moja odpowiedź jest następująca: owszem, wiesz, ale jeśli zechcesz skorzystać ze swojego programu po dłuższej przerwie, to już raczej nie będziesz pamiętał. Drugi,znacznie ważniejszy argument jest taki: sytuacja, w której każdy pisze programy sam dla siebie, jest sytuacją przejściową. Jest to możliwe gdy wszyscy traktują programowanie tylko jako dobrą zabawę, gdyż największym zyskiem z napisania programu jest satysfakcja. Przecież na ogół napisanie i przetestowanie programu rozwiązującego konkretne zadanie trwa dłużej niż rozwiązanie zadania „ręcznie"! Czyli, jeśli mając coś do rozwiązania musimy zacząć od napisania programu, zauważamy rzecz niezwykle przykrą: zastosowanie kosztownego komputera miało nam zaoszczędzić czasu i pracy, a tymczasem stało się wręcz przeciwnie!! Przecież tak być nie może! Dlatego właśnie jest tylko jedno wyjście — trzeba pisać programy tak, aby w przyszłości mogły być wykorzystywane przez wiele osób. Oczywiście sama forma nie decyduje o wszystkim. Pozostaje jeszcze treść. Najuprzejmiejszy program nie wzbudzi mojego zainteresowania jeśli nie da się zastosować do rozwiązywania moich problemów. Ale o tym co należy robić, aby jeden program nadawał się do rozwiązywania wielu podobnych do siebie zadań, porozmawiamy innym razem.

Andrzej Pilaszek

NIE ŚWIĘCI GARNKI LEPIĄ

W poprzednim numerze zaproponowałem Wam stworzenie na własny użytek niewielkiej bazy danych. Jest to zadanie skomplikowane i na jego wykonanie potrzeba sporo czasu, dlatego z opisem możliwych rozwiązań poczekamy jeszcze trochę, żeby nie psuć zabawy tym, którzy dali się namówić. Ponieważ jednak, szczególnie dla mało doświadczonych programistów, zadanie może okazać się bardzo trudne, chciałbym dziś podać kilka ogólnych wskazówek, które mogą nieco ułatwić pracę (jeśli nie chcesz ich czytać, możesz pominąć tekst aż do tytułu „Nowe zadanie").

Zawartość naszej bazy danych musi być przechowywana w pamięci zewnętrznej. W najprostszym przypadku dane w pamięci zewnętrznej stanowią jeden zwarty zapis, który W CAŁOŚCI można wczytać do pamięci operacyjnej (jest to dość poważne ograniczenie rozmiaru bazy danych, ale na początek możemy je zaakceptować). Prace z użytkownikiem rozpoczynamy za każdym razem od wczytania całości bazy do pamięci operacyjnej. Wszystkie operacje na danych — przeszukiwanie, dopisywanie itd. — wykonujemy w tej pamięci. Może to być niezbyt skomplikowane gdy całą zawartość bazy umieścimy w jednej lub kilku tablicach. Po zakończeniu pracy całą zawartość tablic wypisujemy z powrotem do pamięci zewnętrznej, jako jeden niepodzielny zbiór. Będzie tam zapisany do czasu następnego uruchomienia programu.



Minimalny repertuar instrukcji w naszej bazie to:

— wpisanie nowego rekordu,
— wyświetlenie wybranego rekordu (lub grupy rekordów spełniających zadany warunek). Użytkownik musi mieć możliwość wskazania, o które rekordy mu chodzi,
— zmiana wartości wpisanych w wybranym rekordzie, czyli aktualizacja danych,
— usunięcie wybranego rekordu.

Żeby sprawę trochę uprościć możemy się umówić, że nie będzie żadnych dodatkowych mechanizmów wyboru rekordu do aktualizacji lub usunięcia. Wybrana operacja będzie wykonana na tym rekordzie, który ostatnio był wyświetlony. (Uwaga na sytuację, gdy od razu po rozpoczęciu pracy, PRZED wyświetleniem jakiegokolwiek rekordu, użytkownik chce wyrzucać lub aktualizować!).

Oprócz, zapisanych w rekordach danych wpisanych do bazy przez użytkownika, program prawdopodobnie będzie musiał przechowywać (także w pamięci zewnętrznej) własne dane pomocnicze, np. ile rekordów jest już wpisanych, ile zostało jeszcze wolnego miejsca, gdzie to wolne miejsce się znajduje, itd. Dobrze by było gdyby potrafił ponownie wykorzystać miejsce po rekordach, które zostały usunięte z bazy.

Nowe zadanie

Ponieważ zaczął się już rok szkolny proponuję pomyśleć, o zastosowaniu mikrokomputera jako pomocy naukowej. Czy nie uważacie, że fajnie by było napisać program, który uatrakcyjni i ułatwi naukę matematyki. Jako przykład proponuję „skomputeryzować" lekcję pt. „Rozwiązywanie nierówności kwadratowej".

Działanie programu może przebiegać wg następującego schematu:

— wyświetlenie krótkiej informacji o zadaniu („Rozwiązać nierówność kwadratową ax2+bx+c=0),
— wyświetlenie krótkiej informacji o metodzie rozwiązania („Aby rozwiązać nierówność należy obliczyć wyznacznik ze wzoru DELTA = b2-4ac").
— pokazanie na przykładach konkretnych nierówności, których współczynniki są zakodowane w programie, jak przebiega rozwiązanie
— kolejno obliczanie wyznacznika, pierwiastków trójmianu, określanie położenia paraboli i wreszcie wskazanie rozwiązania nierówności. Podstawowe znaczenie ma ilustracja graficzna, a więc wyświetlanie wzorów i obliczeń prowadzących do rozwiązania, ale przede wszystkim interpretacja graficzna zagadnienia, czyli pokazanie rozwiązania na wykresie. Demonstracja powinna uwzględnić wszystkie przypadki, które można napotkać przy rozwiązywaniu zadania a więc wyznacznik dodatni, równy zero, ujemny, współczynnik „a" dodatni i ujemny. Poszczególne przypadki powinny być krótko omówione podczas rozwiązywania. — rozwiązywanie, tak jak w poprzednim punkcie, nierówności o wspóczynnikach podanych przez uczniów. Uwaga: program powinien sprawdzać, czy podane współczynniki rzeczywiście określają nierówność kwadratową.

Tak jak poprzednio, zachęcam tych, którym zadanie się spodoba i zechcą je rozwiązać, do przesłania nam swoich produktów. Wszystkie dobre pomysły mają szanse na opublikowanie w „Bajtku".

(api)

WSZYSTKO DLA WSZYSTKICH

Jak zaprenumerować Bajtka?

Jak reklamować się w Bajtku?

INFORMACJA O CENACH I WARUNKACH PRENUMERATY

Warunki prenumeraty:

1. dla osób prawnych — instytucji i zakładów pracy:

— instytucje i zakłady pracy zlokalizowane w miastach wojewódzkich i pozostałych miastach, w których znajdują się siedziby Oddziałów RSW „Prasa-Ksiaż-ka-Ruch" zamawiają prenumeratę w tych oddziałach;

— instytucje i zakłady pracy zlokalizowane w miejscowościach, gdzie nie ma Oddziałów „Prasa-Książka-Ruch" i na terenach wiejskich opłacają prenumeratę w urzędach pocztowych i u doręczycieli;

2. dla osób fizycznych — indywidualnych prenumeratorów:

— osoby fizyczne zamieszkałe na wsi i w miejscowościach gdzie nie ma Oddziałów RSW „Prasa-Książka-Ruch" opłacają prenumeratę w urzędach pocztowych i u doręczycieli;

— osoby fizyczne zamieszkałe w miastach — siedzibach Oddziałówłów RSW „Prasa-Książka-Ruch", opłacają prenumeratę wyłącznie w urzędach pocztowych nadawczo-oddawczych właściwych dla miejsca zamieszkania prenumeratora. Wpłaty dokonują używając „blankietu wpłaty" na rachunek bankowy miejscowego Oddziału RSW „Prasa-Książka-Ruch".

3. Prenumeratę ze zleceniem wysyłki za granicę przyjmuje RSW „Prasa-Książka-Ruch", Centrala Kolportażu Prasy i Wydawnictw ul. Towarowa 28, 00-958 Warszawa, konto NBP XV Oddział w Warszawie Nr 1163-201045-139-11. Prenu-merata ze zleceniem wysyłki za granicę pocztą zwykłą jest droższa od prenumeraty krajowej o 50% dla zleceniodawców indywidualnych i o 100% dla zlecających instytucji i zakładów pracy.

Termin przyjmowania prenumeraty na kraj i za granicę:

— od dnia 10 listopada na I kwartał, I półrocze roku następnego oraz cały rok następny,

— do dnia l-go każdego miesiąca poprzedzającego okres prenumeraty roku bieżącego.

Uwaga! Za miesiąc opublikujemy druk przekazu pocztowego na prenumeratę „Bajtka" na rok 1987.

INFORMACJA O SPOSOBIE ZAMAWIANIA REKLAM W BAJTKU

Jeżeli chcesz zareklamować się w „Bajtku", zgłoś się do Redakcji Wydawnictw Poradniczych i Reklamy Młodzieżowej Agencji Wydawniczej, Al. Stanów Zjednoczonych 53, 04-028 Warszawa, pokój 313, tel. 13-20-40 do 9 wewn. 403. Cena za 1 cm2 wynosi 200 zł.

Do ceny podstawowej doliczane jest 30% za dodatkowy kolor i 100% w przypadku reklamy wielobarwnej.

Na specjalne życzenie opracowanie reklam powierzamy współpracującym z nami grafikom, fotoreporterom i dziennikarzom. W przypadku indywidualnych zgłoszeń, zlecenia na reklamę prosimy przesyłać na wyżej wymieniony adres, a pieniądze wpłacać bezpośrednio na nasze konto:

NBP XV O/M Warszawa nr 1153-201478-139-11 Młodzieżowa Agencja Wydawnicza Al. Stanów Zjednoczonych 53, 04-028 Warszawa

Zlecenia na reklamę przyjmujemy z sześciotygodniowym wyprzedzeniem.

NIE BÓJ SIĘ MNIE

SZKOŁA NIE Z BAJKI

— Postawiłem sprawę jasno: kto zamierza tu pracować, a jednocześnie czułby się niezręcznie wobec uczniów, którzy opanowali zasady techniki obliczeniowej — powinien zmienić szkołę! Uczyniła to tylko trójka kolegów – Pozostali, czyli absolutna większość spytała: „Kiedy idziemy na kurs?" — wspomina Konstandin Dimitrow.

Miałem więc szczęście spotkać człowieka, który zawodową pasją aż zaraża. Myślę, że takim właśnie cecham charakteru i wiedzą powinien dysponować pedagog wprowadzający swych wychowanków w przyszłe stulecie. **Konstandin Dimitrow** jest dyrektorem 79 Zespolonej Średniej Szkoły Politechnicznej imienia Indiry Gandhi w Sofii.

Niezwykła przygoda komputerowa tej bułgarskiej placówki oświatowej rozpoczęła się w 1983 roku. Wtedy to do „Indiry Gandhi" trafiło siedem maszyn z próbnej serii rodzimej produkcji „INCO-2". Dyrektor szkoły powitał je z radością. Jako wieloletni wcześniej wykładowca Centralnego Instytutu Techniki Obliczeniowej wiedział, że te „myślące urządzenia" mogą wywrócić dotychczasowe metody nauczania. Wtedy też Konstandin Dimitrow zapragnął znaleźć się w ekskluzywnym gronie pionierów, wśród tych, którzy będą przecierać nowe drogi w bułgarskim szkolnictwie. Najważniejsze okazało się, aby

przekonać nauczycieli

Pierwsze komputery dyrektorowi „Indiry Gandhi" pozwoliły zarzucić sieci nie na uczniów, lecz na nauczycieli. Nie u każdego widział on skłonność do przyjęcia wyzwania techniki obliczeniowej. Nie przypadkiem do pierwszej piątki kolegów, który udali się na trzymiesięczny kurs w Instytucie Maszyn Mechaniki Elektronicznej — obok 3 matematyków i fizyka włączył geografa. Chciał w ten sposób udowodnić i przekonać innych szkolnych wykładowców, że komputer nie jest czarną magią, że jego tajemnice potrafi rozgryźć każdy, kto włoży w to trochę wysiłku. Dlatego też pierwszą lekcją z wykorzystaniem komputera nie była matematyka, lecz geografia. Jej temat brzmiał: „Ziemia, planeta układu słonecznego".

W roku szkolnym 1983/84 pierwsza przeszkolona grupa nauczycieli ze wspomnianym geografem zaczęła

zaprzyjaźniać uczniów z komputerami

Obecnie wszyscy z klas od IV do XI potrafią posługiwać się gotowymi programami.

— **Zawsze uważałem** — notuję kolejną wypowiedź szefa „Indiry Gandhi" — **że osobisty przykład nauczyciela ma ogromne znaczenie w upowszechnianiu techniki obliczeniowej. Każdy, kto kieruje tu podanie o pracę, musi mieć świadomość czekającego go wysiłku. Jak mówi Todor Żiwkow, jeśli pragniemy coś zostawić przyszłemu pokoleniu, to musimy żyć i działać w przeświadczeniu o niezwykłości naszego zawodu, który jest rodzajem powołania.**

Placówka kierowana przez **Konstandina Dimitrowa** zdobyła mocną pozycję na oświatowej mapie Soffi. Za wzorową pracę nauczyciele otrzymują liczne wyróżnienia — nagrody z resortu oświaty, władz miejskich i dzielnicowych, związków zawodowych. Do dyplomów często dołączane są koperty z lewami. Niedawno poza pieniężnymi nagrodami indywidualnymi cały kolektyw otrzymał 1500 lewów.

W tej szkole aż chce się pracować. Dodatkowym bodźcem — przestrzegana zasada: „dobra praca — dobra płaca".

W „Indirze Gandhi" uczy się blisko 2 tysiące dziewcząt i chłopców. Od połowy 1985 roku korzystają oni z 55 komputerów osobistych „Prawec 82", zamontowanych w trzech oddzielnych salach.

Codziennie od ósmej rano do siedemnastej szkolne komputery nieprzerwanie pracują. Choć są wakacie przed ekranami siedzi wielu uczniów. Od Marii Stoiłowej z VII „b" słyszę, że rówieśnicy z innych szkół zazdroszczą jej chodzenia do „Indiry Gandhi". Znajdującego się przy sąsiednim stanowisku Georgi Iwanowa ponoć siłą trzeba odrywać od klawiatury, chce zostać elektronikiem. Pełniąca dyżur **Irina Tankowa**, matematyczka, należy do ścisłego grona wykładowców, którzy jako pierwsi złapali w żagle komputerowy wiatr. Twierdzi, ze dzięki tym nowym metodom nauczania uczniowie nieporównanie szybciej i lepiej przyswajają materiał ze wszystkich niemal przedmiotów.

W odrębnym pomieszczeniu stoi

mózg szkoły

— komputer **BMC**. Do jego pamięci nauczyciele wprowadzają informacje o każdej klasie. Bez wertowania dzienników natychmiast można poznać sytuację danej grupy uczniów. W nowym roku szkolnym również rodzicom udostępni się ten wszystko wiedzący o ich pociechach ekran.

Matematyczka **Mitica Nikołowa** demonstruje mi następnie sposób bieżącego kontrolowania pracy każdego podopiecznego bèz ruszania się z profesorskiej katedry. Również uczeń bez wstawania może zajrzeć na ekran wykładowcy — sprzężenie zwrotne.

Mój cicerone — dyrektor **Dimitrow** niejako na deser zostawia szkoleniową linię automatyczną, złożoną z minifrezarki, minitokarki, lasera i zaplecza magazynowego z detalami do obróbki. Robot wykonuje komendy, wydawane przez odpowiednio zaprogramowany komputer. Czy można sobie wyobrazić cudowniejszą możliwość praktycznych ćwiczeń dla techników przyszłości? Na powstające tu elementy czeka zakład przemysłu optycznego. Szkolny projekt, szkolne wdrożenie?.

— **Obecnie na koncie mamy milion lewów** — akcentuj Konstadin Dimitrow. — **Jak człowiek szuka to znajdzie, a najważniejsze to chcieć zrobić! Różne zjednoczenia, radę dzielnicową, Dimitrowski Komsomoł, ministerstwa finansów i oświaty, przekonaliśmy, że w naszych wychowanków warto inwestować. Tylko tacy jak oni są w stanie podjąć trud realizacji zadań narodowego programu rewolucji naukowo-technicznej.**

— **Sami zresztą też już zarabiamy jako współwykonawcy rynkowych wyrobów optycznych. Ponadto trwa budowa szkolnego centrum obliczeniowego, które będzie obsługiwało dzielnicowe przedsiębiorstwa, zakłady i instytucje. Naszej szkole marzy się samowystarczalność, co jest bardzo realne i bliskie czasowo.**

Język „BASIC" i komputery nie zdominowały procesu dydaktycznego, choć są to jego główne elementy. Szkoła posiada rozwiniętą bazę do uprawiania różnych dyscyplin sportowych. W prawidłowym i wszechstronnym rozwoju osobowości uczniów „Indiry Gandhi" sprawność fizyczna odgrywa istotną rolę. Zacieśniają się kontakty szkoły z sofijskim Domem Literatury Dzieci i Młodzieży. Twórcy i działacze kultury przebywają w szkolnych murach już nie na zasadzie gości.

Absolwenci powinni być ludźmi znającymi się nie tylko na komputerach, rozległa wiedza i zainteresowania pozwalają te myślące maszyny lepiej wykorzystywać, gdyż komputer to nie cel, lecz środek sposób i metoda rozwiązywania złożonych problemów.

Od **Iwana Cwiełanowa**, głównego specjalisty w Głównym Centrum informatycznym Ministerstwa Oświaty dowiaduję się, że 79 Zespolona Średnia Szkoła Politechniczna im. Indiry Gandhi w Sofii nie jest pokazową i jedyną tego rodzaju placówką w Bułgarii. W końcu roku we wszystkich gimnazjach i „zawodówkach" znajdzie się minimum po 9 komputerów. Plany na rok 1987 są jeszcze ambitniejsze.

Grzegorz Lubczyk





ABAKUS I HCC

Stowarzyszenie Mikrokomputerowe **ABAKUS** gościło w lipcu br. Przedstawiciela **Hobby Komputer Club** z Holandii, pana Macieja Kolio. Nasz gość jest członkiem zarządu **HCC** odpowiedzialnym za współpracę z grupami zainteresowań w kraju i za granicą. Przedstawiciele **ABAKUSA** przeprowadzili rozmowy wstępne na temat możliwości współpracy obu organizacji 26 lipca p. M. Kolio został zaproszony na spotkanie zorganizowane w siedzibie **ABAKUSA** w „Warszawskim Ośrodku Kultury przy ul. Elektoralnej 12, podczas którego przedstawił w skrócie system organizacyjny **HCC**. W spotkaniu wzięli również udział, oprócz członków stowarzy szenia, przedstawiciele „Bajtka" i „Komputera".

HCC jest największą — bo liczącą 30 tysięcy członków — organizacją społeczną skupiającą mikrokomputerowych sympatyków. Zarząd **HCC** liczy ok. 200 osób – przedstawicieli wszystkich grup zainteresowań. Zarząd ocenia plany pracy klubów i ich realizację i przyznaje w razie potrzeb pożyczki potrzebne do realizacji programów — spłacane później w wieloletnich ratach. **HCC** dysponuje między innymi działającą w całej Holandii siecią mikrokomputerową — **FIDO**. Przy pomocy mikrokomputera podłączonego poprzez modem do telefonu domowego, można komunikować się z centralną bazą danych i innym członkam **HCC** Stowarzyszenie wydaje również własny miesięcznik. Każdy członek otrzymuje bezpłatnie jeden egzemplarz.

Czy **ABAKUS** podejmie współpracę z HCC?

Taka współpraca jest pożądana. **HCC** powstał w 1976 roku, gdy o mikrokomputerach domowych marzyli tylko fantaści. Ma więc wieloletnie doświadczenie w pracy społecznej i integracji tak wielkiej rzeszy ludzi.

W **HCC** większość członków pracuje na własnym sprzęcie. Prywatne mikrokomputery są powierzane klubowi w miarę potrzeb i na nich pracują zainteresowani członkowie sekcji. Czy u nas jest to możliwe? Chyba tak — ale pod warunkiem wykorzystywania tego sprzętu tylko do obliczeń na gotowych programach lub do ich tworzenia. Trzeba również podkreślić, że sprzęt Stowarzyszenia powinien mieć umożliwiony tani serwis. W tym celu musi powstać w **ABAKUSIE** silna sekcja hardwareowa. Zaoszczędzilibyśmy dziesiątki tysięcy złotych na naprawy. Ponadto można będzie nawiązać współpracę z krajowymi producentami nielicznych części zamiennych. Sekcja sprzętowa z czasem może przekształcić się w sekcję amatorskiej budowy mikrokomputerów.

HCC współpracuje z wieloma firmami, które powierzają klubowi produkowany przez nie sprzęt do testowania i oceny. Wyniki testów są publikowane na łamach własnego wydawnictwa i jedna z ważniejszych form współpracy, to wymiana oprogramowania między odpowiednimi sekcjami zainteresowań i tworzenie programów na konkretne zamówienia. *(as)*

NOTESIK

Cześć Maluchy!

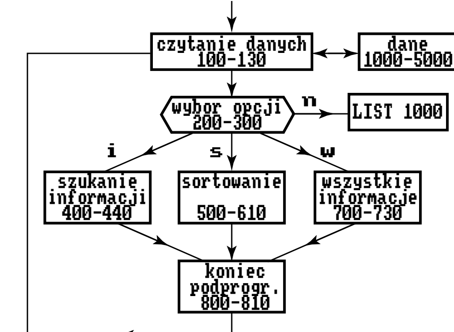
Do dzisiaj wszystkie pisane przez nas programy służyły tylko do zabawy. Tym razem zabierzemy się za prawdziwy program użytkowy. Co prawda dość prosty, ale zawsze...

Chyba każdy przedszkolak, który bawił się komputerem wie, co to jest baza danych. Jest to po prostu taki komputerowy notes, który sam — na twoje żądanie — odszukuje potrzebne ci informację. Powiedzmy, że chcemy zrobić sobie taką bazę danych z numerami telefonów naszych kolegów. Zastanówmy się co chcielibyśmy by ten program zrobił.

Po pierwsze musi on pamiętać pewną liczbę nazwisk, imion i numerów telefonów. I tu warto zaglądnać do jednego z poprzednich odcinków „Tylko dla przedszkolaków” — „Skarb Kubusia” („Bajtek” nr 7/86). Mówiliśmy tam o tablicach, które będą niezbędne przy konstruowaniu naszej bazy danych. Stworzymy sobie więc tablicę do przechowywania zmiennych łańcuchowych — czyli tekstów. Nazwijmy ją: notes$ (znaczek $ na końcu mówi, że jest to tablica do przechowywania tekstów).

W linii 100 naszego programu umieściliśmy deklarację tej tablicy. Liczba 1000 określa, ilu kolegów będziemy mogli zapisać w naszym komputerowym notesie (macie ich tylu?), liczba 2 natomiast mówi ile zmieści się informacji o każdym z nich. Oczywiście chcemy wpisać imię, nazwisko i numer telefonu. Więc trzy informacje a nie dwie. Pomyłka? Nie, wszystko w porządku. Po prostu imię będzie miało numer 0 — notes$(poz,0), nazwisko numer 1 — notes$(poz,1) a telefon będzie miał numer 2 — notes$(poz,2). Jak już się pewnie sami domyśliliście zmienna: poz. oznacza pozycję kolegi w naszym notesie.

Przechowywanie danych w tablicach — oprócz wielu zalet — ma również swoje wady. Nie da się ich zapisać wraz z programem na taśmę. W „prawdziwych” bazach danych informacje do wypełniania tablic przechowywane są jako osobne zbiory na dyskach czy taśmach. Po wczytaniu programu bazy wczytujemy następnie dane. Jeśli uzupełnimy naszą bazę o nową informację, musimy dane do naszej bazy nagrać ponownie.

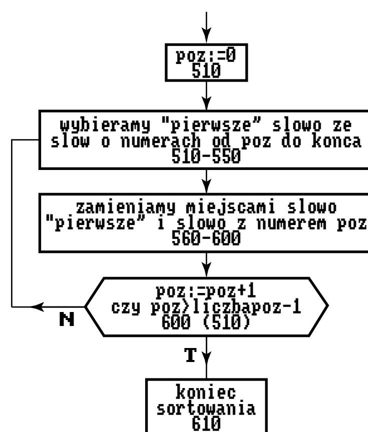


Rys. 1. Schemat blokowy dzialania programu bazy danych.

Taki sposób pracy jest bardzo wygodny dla posiadaczy stacji dyskietek, czy też twardych dysków. Jest natomiast bardzo uciążliwy dla przedszkolaków, którzy korzystają ze zwykłego magnetofonu. Dlatego też, proponuje inne rozwiązanie. Nowych kolegów wpisujemy bezpośrednio do programu, za instrukcja DATA w następującej kolejności: imię, nazwisko, telefon. Jeśli nie znamy np. nazwiska kolegi, stawiamy w tym miejscu znak „”, co oznacza słowo puste. Po uzupełnieniu danych nagrywamy od nowa cały program, co zważywszy, że jest on dość krótki nie sprawia nam większych kłopotów.

Zasadę działania programu poznamy najlepiej posługując się schematem blokowym przedstawionym na rys. 1. Zwróćcie uwagę, że nie zawiera on wszystkich instrukcji programu, a jedynie pokazuje zależności pomiędzy jego poszczególnymi częściami.

Po uruchomiemu następuje deklaracja tablic i przepisywaniu danych umieszczonych pod instrukcją DATA do tablicy notes$(100-130). Następnie drukowane są możliwości wyboru i odczytywana jest nasza odpowiedź (200-300). W zależności naszej decyzji komputer wykonuje skok do jednego z trzech podprogramów lub — jeśli stwierdziliśmy, ze chcemy wprowadzić nowe dane — wydrukuje nam na ekranie fragment programu, zawierający instrukcję DATA (możemy dopisać wówczas do programu nowych kolegów).



Rys. 2. Schemat blokowy dzialania podprogramu sortowania

Gdy chcemy znaleźć telefon konkretnego kolegi, wybieramy funkcję „szukania informacji” (400-440). Jest ona zrealizowana w bardzo prosty sposób. Najpierw komputer pyta cię czy chcesz podać imię czy nazwisko kolegi, następnie podane przez ciebie słowo zapamiętuje pod zmienną haslo$ i po kolei porównuje ze słowami w tablicy. Jeśli trafi na takie samo słowo drukuje pełną informację, jeżeli takich słów jest więcej drukuje wszystkie. Możesz także zażyczyć sobie wypisania wszystkich danych po kolei (700-730). Komputer wydrukuje je w takiej kolejności w jakiej zostały wczytane z instrukcji DATA, chyba że wcześniej zostały ustawione w kolejności alfabetycznej nazwisk.

Podprogram sortowania (500-610) jest chyba najciekawszym fragmentem naszej bazy danych. Algorytm (sposób) jego działania przedstawia rys. 2. Na początek bierzemy wszystkie nazwiska od pierwszego (z numerem 0) do ostatniego i wybieramy z tej grupy nazwisko „najmniejsze”, czyli pierwsze w kolejności alfabetycznej. Następnie zamieniamy miejscami nazwisko o numerze 0 z nazwiskiem „najmniejszym” i w ten sposób mamy już pierwsze nazwisko na początku. Teraz bierzemy nazwiska od numeru 1 do końca, znów wybieramy „najmniejsze” i zamieniamy je miejscami z nazwiskiem o numerze 1. Drugie nazwisko w kolejności alfabetu jest już na drugim miejscu itd... Powtarzamy te operacje aż zostanie nam tylko jedno słowo.

```
99   REM ****** czytanie danych
     ******
100 DIM notes$(1000,2): DIM
     pierwszy$(2):LET poz=0
110 FOR j=0 TO 2: READ notes$(poz,
     j):IF notes$(poz,j)="koniec"
     THEN GOTO 130
120 NEXT j: LET poz=poz+1: GOTO 110
130 LET liczbapoz=poz-1
199 REM ****** wybieranie opcji
     ******
200 PRINT "      wybieraj:":PRINT
210 PRINT "szukanie informacji (i)"
220 PRINT "wszystkie informacje (w)"
230 PRINT "sortowanie wg. nazwisk
     (s)"
240 PRINT "nowe dane (n)"
250 INPUT odp$
260 IF odp$="i" THEN GOTO 400
270 IF odp$="w" THEN GOTO 700
280 IF odp$="s" THEN GOTO 500
290 IF odp$="n" THEN LIST 1000-
300 GOTO 250
399 REM ****** szukanie informacji
     ******
400 INPUT "Wybierasz wg. nazwiska
     (1) czy imienia 0)";informacja
410 INPUT "Szukane haslo"; haslo$
420 FOR poz=0 TO liczbapoz
430 IF notes$(poz,informacja)=
     haslo$ THEN FOR i=0 TO 2:  PRINT
     notes$(poz,i):NEXT i: PRINT
440 NEXT poz: GOTO 800
499 REM ****** sortowanie ******
500 PRINT "Sortowanie...";
510 FOR poz=0 TO liczbapoz-1
520 LET pierwszy$=notes$(poz,1): LET
     pierwszynr=poz
530 FOR i=poz+1 TO liczbapoz
540 IF pierwszy$>notes$(i,1) THEN
     LET pierwszy$=notes$(i,1): LET
     pierwszynr=i
550 NEXT i
560 IF pierwszynr=poz THEN GOTO 600
570 FOR j=0 TO 2: LET onotes$(j)=no
     tes$(pierwszynr,j): NEXT j
580 FOR j=0 TO 2: LET notes$(pierwsz
     ynr,j)=notes$(poz,j):NEXT j
590 FOR j=0 TO 2: LET notes$(poz,j)=
     onotes$(j): NEXT j
600 NEXT poz
610 PRINT "...koniec": GOTO 810
699 REM ****** wszystkie informacje
     ******
700 PRINT "ENTER - kolejna pozycja"
710 FOR poz=0 TO liczbapoz
720 FOR i=0 TO 2: PRINT
notes$(poz,i), :NEXT i: INPUT;o$:
PRINT
730 NEXT poz
799 REM ****** zakoczenia
     podprogramow ******
800 PRINT "koniec danych"
810 PRINT: PRINT: GOTO 200
999 REM ****** dane ******
1000 DATA "Bajtek","","21-12-05"
1010 DATA "Zyzio", "Bulba", "21"
1020 DATA "Tomcio", "Paluch", "777-
      77"
1030 DATA "Smok", "Wawelski", "13-
      13-13"
1040 DATA "Tomek", "Sawyer",
      "12-34-56"
1050 DATA "Adam","","44-77-44"
1060 DATA "Faust","try2emu",
      "t2e.pl"
5000 DATA "koniec"
```

Warto jeszcze przez chwilę zatrzymać się przy tym podprogramie. Mimo dość prostego algorytmu zawiera on kilka pętli, które powodują, że czas jego wykonania wydłuża się. Jeśli w naszej bazie będziemy mieli wpisane kilku czy kilkunastu kolegów, nie ma to większego znaczenia. Gdy jednak wpiszemy stu, wówczas będziemy musieli poczekać dobrą chwilę aż komputer wykona swoje zadanie. W dużych bazach danych czas odszukiwania i porządkowania informacji ma ogromne znaczenie po prostu nie można pozwolić, by komputer godzinami wykonywał nasze polecenia. Należy więc tak konstruować programy, by osiągnąć zamierzony efekt możliwie najkrótszą drogą i w możliwie najkrótszym czasie. Informatycy nazywają to optymalizacją programu. Okazuje się jednak ze problem dotyczy nie tylko „prawdziwych” programów, pisanych przez „prawdziwych” programistów, ale także i twoich. Zastanów się nad tym planując swój następny program.

Romek

GIEŁDA

		BAJTKOWA GIEŁDA (tys. zl)	KOMIS (tys. zł)	BOMIS (tys. zł)	BALTONA REWEX (Dol. USA)	RFN (średnie) (DM)	WLK. BRYT. (średnie) (£)
SINCLAIR	**ZX81**	25–30	25-40	–	–	90	–
	ZX Spectrum 48 kB	70–80	105-110	130	125	150-250	45–65
	ZX Spectrum Plus	100–115	140-160	180	–	200-340	70–90
	ZX Spectrum 128 kB	195	–	–	–	520-590	150
	Drukarka SEIKOSHA GP50S	85-90	100	150	–	260	60–65
	Interface Kempston	8-16	20	–	–	40	7-9
	Magneton (średn. jak.)	20-35	40-45	40-60	–	50-120	15-20
COMMODORE	**C–64**	130–150	150–160	240	–	450–520	90–110
	C–128	280–350	450	450–500	–	700	210–230
	C–128D	600	–	–	–	1500	390-410
	Magnetofon 1531	20–30	35	–	–	80–110	–
	Stacja dyskietek 1541	150–160	200	350	–	520	110–150
	Stacja dyskietek 1570	200	-	700	–	490–540	160–170
	Drukarka MPS 801	120–130	150	280	–	199	–
	Drukarka MPS 803	–	-	360	–	300-330	140–150
	Dyskietki 51/4 (średnia jak.)	0.7–1.8	1.2 – 1.5	2–3.5	–	1.2–5	0.8–2
ATARI	**800 XL**	70–80	90	–	115	140-180	70–80
	130 XE	140-160	190	380	199	360	150
	Stacja dyskietek 1050	130–140	200	–	185	478	130
	Magnetofon	25–30	35	-	48	75	20
	Drukarka 1029	140	200	–	199	–	85
AMSTRAD	**464 z mon. monochromat.**	190-220	300	500	–	680–720	160–180
	464 z monitorem kolor.	290	350	600	–	1100–1200	260–268
	664 z mon. monochromat.	300-350	–	600–700	–	1000–1300	–
	6128 z mon. monochromat.	370-400	700	850	–	1450–1500	250–260
	6128 z mon. kolor.	455	–	1.1 min.	–	1900–2000	340–350
	PCW 8256	900	1 min.	1.2 min.	–	2000–2100	380–400
	Dyskietki 3"	3.5–5	5–7	8	6	10-15	3.5–4

SPRZĘŻENIE

Drogi

Jak działa „myszka"? Czy można „myszkę" podłączyć do Commodore 64? Czy pióro świetlne podłączone do mikrokomputera działa na ekranie telewizora, czy tylko na ekranie monitora?

Agnieszka Szkatuła
ul. 1 Maja 133
44-340 Skrzynów

W dolnej części myszki, stykającej się z podłożem, znajduje się kula, która obraca się, gdy przesuwamy „myszkę" po stole. Komputer odczytuje ruch kulki i stosownie do tego przesuwa kursor po ekranie. Ma górze „myszki" znajdują się jeden lub dwa przyciski — mikroprzełączniki. Ich funkcja zależy od konkretnego programu.

Do Commodore 64 jak również i 128 można podłączyć „myszkę". Istnieje kilka wersji tych urządzeń produkowanych przez różne firmy. Jest także dostępne oprogramowanie do tego urządzenia.

Pióro świetlne działa zarówno z monitorem jak i ze zwykłym telewizorem

Właśnie otrzymałem w prezencie mikrokomputer Commodore 64. Proszę o podanie kilku tytułów czasopism zachodnich o tym komputerze (znam trochę język angielski).

JB
Wrocław

Oto najpopularniejsze pisma dotyczące Twojego komputera, wydawane w języku angielskim:

Your Commodore (GB)
Commodore Computing International (GB)
Commodore Microcomputers (USA)

Posiadam komputer ATARI800 XL. Zrobiłem opisane w Waszym piśmie wiosełka, ale nie mam żadnego programu, w którym są one wykorzystywane.

Adam Ryjski
Żelechów

Możesz wykorzystywać je we własnych programach. ATARI BASIC posiada instrukcję PADDLE (wyrażenie numeryczne), która przyporządkowuje liczbę położeniu wskazanego potencjometra „wiosełek". Stosując tę instrukcję możesz napisać np. ciekawy program graficzny.

W numerze 3-4/86 „Bajtka", na stronie 22 w programie „Niespodzianka" wkradł się chyba błąd...

Krzysztof Wieczorek
ul. Grunwaldzka 116/19
80-244 Gdańsk

Niestety to prawda. W linii 40. po instrukcji AT złośliwy chochlik drukarski zamienił przecinek pomiędzy liczbami 0 i 26 na kropkę. Linia ta powinna wyglądać tak:

40 PRINT AT 0,26; „Rys":"-";n

Bajtku!

Czy ATARI XE jest sprzedawany z „myszką"? Czy 130 XE może współpracować z firmowym magnetofonem?

Paweł Łęgorz ul. Pleszyńska 15m 10 42-200 Częstochowa Niestety, sprzedawany przez Pewex ATARI 130 XE nie jest wyposażony w myszkę. Może natomiast, bez problemów współpracować z firmowym magnetofonem Atari.

Niedawno kupiłem mikrokomputer Amstrad CPC 6128. Proszę o podanie adresów klubów posiadających komputery Amstrad

Michał Banas
ul. Klaudyny 28/82
01-684 Warszawa

Oto trzy wybrane:
1. Klub AMSTUDIO ul. Fredry 7 60-967 Poznań
2. Klub Amstrad ul. Blokowa 1 03-641 Warszawa
3. Klub OMIC
ul. Brzozowa 20a 43-100 Tychy

Jestem Waszym stałym czytelnikiem. Posiadam ATARI 800 XL i magnetofon firmowy. Niestety mieszkam na wsi i mimo usilnych starań nie mogę na wiązać kontaktu z innymi użytkownikami tego typu komputera. Jestem inwalidą I grupy na stałe przykutym do wózka inwalidzkiego, co uniemożliwia mi wyjazdy na giełdy komputerowe i spotkania z innymi użytkownikami. Posiadam już pewne — choć jeszcze niewielkie — doświadczenie w obsłudze i programowaniu. Mam również trochę programów do wymiany. Obiecuję odpisać na każdy otrzymany list, gdyż posiadam dużo wolnego czasu. Chciałbym nawiązać stałą korespondencję z czytelnikami Waszego pisma.

Krzysztof Profus
ul. Ogrodowa 22
42-674 Zbrosławice

Sądzę, że wśród czytelników „Bajtka" znajdzie Pan wielu przyjaciół.

Marcin



Na listy czytelników odpowiada Marcin Waligórski Student III roku Informatyki

HOMO INTELLIGENS

Dokończenie ze str. 32

Trwająca nieprzerwanie prawie 35 tys. lat epoka homo sapiens dobiega: końca. Jesteśmy świadkami narodzin nowego typu człowieka, który nazwany został homo intelligens. Tak przynajmniej twierdzi prof. Yoneji Masudo, jeden z pionierów komputeryzacji w Japonii, autor książki „Information Society".

A wszystkiemu winne są podobno komputery. To one w powiązaniu z nowymi technikami przekazu informacji i robotyzacji stać się mają bazą materialną tego przełomu. Komputer, według, profesora Yoneji Masudo, jest trzecim kolejnym najdonośleJszym wynalazkiem człowieka. Pierwszym była siekiera krzemienna, drugim maszyna parowa, którą dała początek rewolucji przemysłowej, trzecim jest komputer.

A zaczęło się to tak niewinnie, w 1946 r. Od tego czasu w życie weszły kolejne generacje komputerów. W 1983 r. przystąpiono już do budowy piątej generacji komputerów, a naukowcom, śni się prawdziwy cud techniki. Komputer szóstej generacji wyposażony w strukturę i funkcje podobne do umysłu ludzkiego, sztuczny mózg, biokomputer rozszerzy znacznie możliwości ludzkiego rozumu. Nie będzie go tylko wyręczał w mozolnych pracach, przyspieszał operacji. Będzie stanowił naturalne przedłużenie ludzkiego intelektu.

Czy rewolucja, jaką z sobą niesie komputer może oddziałać na samego człowieka, zmienić go, nauczyć inaczej myśleć, inaczej pracować, inaczej organizować życie społeczne?

Każdy przełomowy wynalazek w historii zmieniał co nieco w życiu człowieka i społeczeństw. Ale tym razem oczekuje się zmiany radykalnej, epokowej. Jeśli w rozwoju człowieka decydujące znaczenie miało opanowanie mowy, doskonalenie ręki, a tym samym umiejętności posługiwania się narzędziem, a także rozwój mózgu, jakież możliwości kryć będzie teraz to nowe sprzężenie mózgu ludzkiego i mózgu biokomputera.

Czy komputer nauczy się ludzkiego przekazywania informacji? Wiadomo, że zaledwie kilkanaście procent przekazywanej informacji między ludźmi zawarte jest w słowach. Reszta przypada na grymasy, mimikę, intonację, gesty i ruchy. A może będzie odwrotnie? Konieczne będzie nauczenie się precyzyjnego myślenia i przekazywania informacji.

Na genezę homo intelligens mają złożyć się następujące fakty: ewolucja człowieka, wprowadzenie i upowszechnienie piątej i szóstej generacji komputerów, rozwój nowych środków łączności i komunikowania się, zaawansowana robotyzacja i ko-ewolucja genetyczno-kulturalna.

Jaka będzie cywilizacja, która powstanie w wyniku zbudowania sieci informacyjnej, przystosowanej do indywidualnego odbiorcy i autonomicznie przez niego formowanej. Jakie będą jej cele, kiedy dalszy rozwój nie będzie dyktowany potrzebą pomnażania dóbr materialnych? Pomnażanie informacji różni się bowiem zdecydowanie od pomnażania dóbr. Możliwości pomnażania informacji są wprost nieograniczone, a zasoby materialne mają to do siebie, że podlegają wyczerpaniu. Przy wymianie dóbr jeden traci drugi zyskuje. A przy przekazywaniu informacji pozostaje ona także u tego, od kogo pochodzi. Ponadto informacja stwarza niespotykane możliwości jej kumulacji, a w przypadku powiększenia rzeczy materialnych uzyskany efekt może być odwrotny od zamierzonego.

A efekty?

Czy zdajemy sobie sprawę, jaką rewolucję pociągnie za sobą powstanie cywilizacji sieci informacyjnej, przystosowanej do indywidualnego odbiorcy i autonomicznie przez niego kształtowanej, której bazą techniczną będą komputery osobiste, łączność satelitarna i szereg innych systemów.

Czy ta nowa technika ułatwi zrozumienie z natury złożonych problemów współczesnego świata i rozwiązanie takich globalnych kwestii jak eksplozja demograficzna, zanieczyszczenie środowiska, zapobieżenie wojnie termojądrowej.

Profesor Yoneji Masudo uważa, że jest to możliwe. Jego zdaniem świat zmierza w kierunku nowego modelu organizacji. Będzie on opierał się na gospodarce bioekologicznej, zakładającej ograniczoną konsumpcję, wielokrotne wykorzystywanie zasobów i synergię pomiędzy produkcją i sposobem jej wykorzystania.

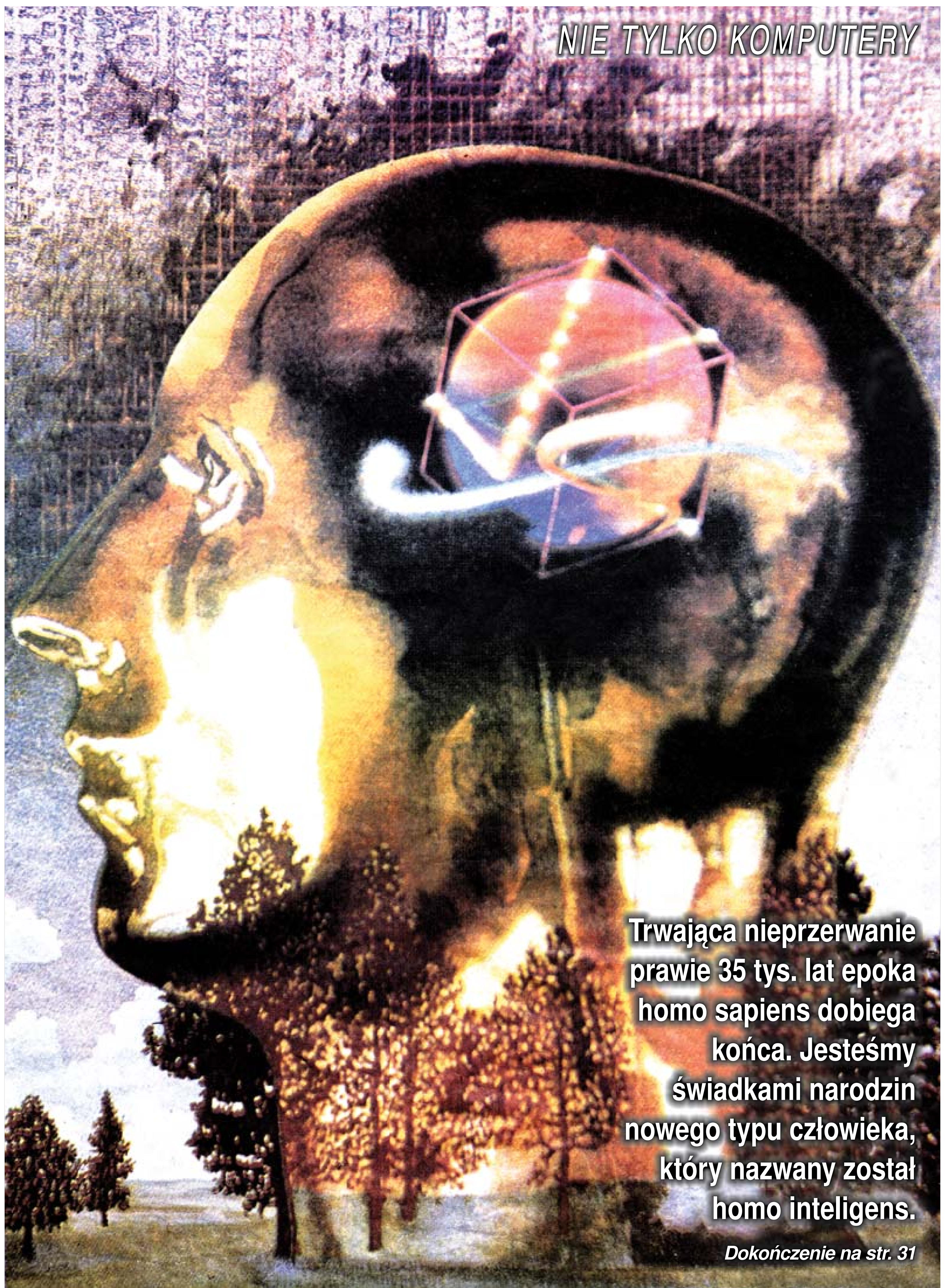
Czy nowa technika z jej ogromnymi możliwościami w zakresie „doinformowania się", zrozumienia i porozumienia ułatwi rozwiązanie palących problemów współczesności, z których część nabrała już wymiaru globalnego, jak np. eksplozja demograficzna, zagrożenie środowiska.

Czy jest szansa na realizację wizji prof. Yoneji Masudo, który uważa, że przyszłość świata należeć będzie do homo intelligens i do społeczeństwa synergicznego, wzajemnie sobie pomagającego, opartego na bioekologicznym systemie ograniczonej konsumpcji, wielokrotnego wykorzystania zasobów i synergii między produkcją i jej wykorzystaniem.

Sama techniką, mimo jej ogromnego oddzia ływania nie jest w stanie zmienić człowieka. Ale też nowoczesna technika stwarza coraz większe możliwości dla kreatywnego oddziaływania na rzeczywistość. Stwarza szansę na tworzenie pożądanego świata i wyrwanie się z zaklętego kręgu świata konieczności.

(ak)

HOMO INTELIGENS



NIE TYLKO KOMPUTERY

Trwająca nieprzerwanie prawie 35 tys. lat epoka homo sapiens dobiega końca. Jesteśmy świadkami narodzin nowego typu człowieka, który nazwany został homo inteligens.

Dokończenie na str. 31



www.t2e.pl

Seria „Reduks" ma na celu przetworzenie – pełne zdigitalizowanie prasy komputerowej z początków okresu informatyzacji naszego kraju.

W odróżnieniu od plików w formacie djvu i cbr gdzie format papierowy jest tylko zeskanowany i połączony w jeden dokument, **reduks polega na ponownym przełamaniu dokumentu, wraz z ponowną obróbką graficzną - często z ich ponowną kreacją**. Tam gdzie to możliwe zachowana jest grafika z pierwowzoru, jednak jeśli materiał jest zbyt złej jakości - zostaje on podmieniony (oczywiście jeśli jest dostępny) na równoważny. Jeżeli uda się odnaleźć błąd w zamieszczonych listingach są one poprawiane tak, aby mogły zostać uruchomione na emulatorach lub mikrokomputerach.

Rozpocząłem ten cykl od „Bajtków", jednak mam nadzieję, że uda mi się wzbogacić kolekcję zdigitalizowanych i ponownie przełamanych periodyków także o inne ważne tytuły z lat ubiegłych, o których powoli zapominamy, a bez których współczesny obraz komputeryzacji wyglądałby zgoła inaczej.

Wydania „Reduksowe" są bezpłatne, nie roszczę sobie żadnych praw do zawartości przetwarzanych magazynów, jednak ilość czasu i pracy jaką wkładam aby mogły one trafić w Twoje ręce jest na tyle spora, że pozwolę sobie napomknąć iż możesz wesprzeć mnie przekazując dotację w dowolnej wysokości.

Jeśli posiadasz materiały fotograficzne lub edytorskie, które są lepszej jakości niż prezentowane w numerach, daj znać.



W ramach wolnego czasu odtwarzam stare magazyny komputerowe i prowadzę serwis emulacyjny try2emu. Jeżeli uważasz, że moja publiczna działalność możesz postawić mi kawę, a kawę lubię;)



https://buycoffee.to/t2e

Osoby, które wspierają serwis datkami* mają dostęp do reduksów w wysokiej rozdzielczości z usuniętymi reklamami.

W zależności od wysokości datku każdy otrzymuje czasowy dostęp do reduksów w pełnej rozdzielczości na miesiąc, 3 miesiące lub 6 miesięcy.
Osoby, które aktywnie wspierają projekt mają dostęp nieograniczony.